

AD-A058 632

MITRE CORP BEDFORD MASS

SEEK IGLOO LIFE CYCLE COST MODEL. VOLUME III. MAINTENANCE MANUA--ETC(U)

JUL 78 J K FERRAILOLO

F19628-78-C-0001

UNCLASSIFIED

MTR-3577-VOL-3

ESD-TR-78-155-VOL-3

NL

1 OF 2
AD
A058632



LEVEL *11*

(12)

g

ESD-TR-78-155, Vol. III

MTR-3577, Vol. III

AD A058632

SEEK IGLOO LIFE CYCLE COST MODEL
MAINTENANCE MANUAL

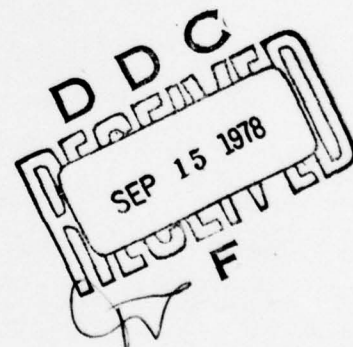
BY J. K. FERRAILO

JULY 1978

Prepared for

DEPUTY FOR SURVEILLANCE AND NAVIGATION SYSTEMS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Massachusetts

AD No. _____
DDC FILE COPY



Approved for public release;
distribution unlimited.

Project No. 6260

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts

Contract No. F19628-78-C-0001

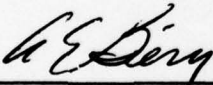
78 09 11 003

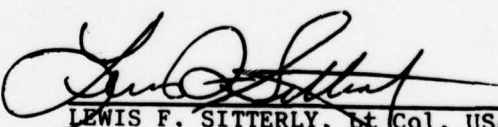
When U.S. Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Do not return this copy. Retain or destroy.

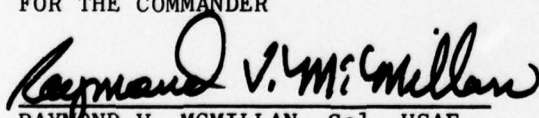
REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.


ALVIN E. BIERY, Major, USAF
Project Officer


LEWIS F. SITTERLY, Lt Col, USAF
Deputy Program Director for SEEK IGLOO

FOR THE COMMANDER


RAYMOND V. MCMILLAN, Col, USAF
Director, North American Airspace
Surveillance Systems Program Office
Deputy for Surveillance and Navigation Systems

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER 18 ESD-TR-78-155 Vol. III - 3	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9 Technical rept.	
4. TITLE (and Subtitle) 6 SEEK IGLOO LIFE CYCLE COST MODEL, Volume III, MAINTENANCE MANUAL.	5. TYPE OF REPORT & PERIOD COVERED		
7. AUTHOR(s) 10 J.K. Ferraiolo	14. PERFORMING ORG. REPORT NUMBER 14 MTR-3577-Vol. III - 3	8. CONTRACT OR GRANT NUMBER(s) 15 F19628-78-C-0001	
9. PERFORMING ORGANIZATION NAME AND ADDRESS The MITRE Corporation P.O. Box 208 Bedford, MA 01730		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project No. 6260	
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Surveillance and Navigation Systems Electronic Systems Division, AFSC Hanscom Air Force Base, MA 01731		12. REPORT DATE 11 JUL 1978	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 157 p.		13. NUMBER OF PAGES 155	
		15. SECURITY CLASS. (of this report) UNCLASSIFIED	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
18. SUPPLEMENTARY NOTES			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) ANALYTIC ACCOUNTING MODEL RADAR LOGISTICS MODEL FORTRAN PROGRAM CODE SEEK IGLOO INTERACTIVE COMPUTER MODEL SENSITIVITY ANALYSIS LIFE CYCLE COST			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) An interactive Life Cycle Cost (LCC) mathematical model with a built-in sensitivity analysis capability has been developed for use in the evaluation of proposed designs for the Air Force SEEK IGLOO Radar System. This Maintenance Manual provides the information necessary to maintain, or possibly modify, the FORTRAN code of the LCC Model. It contains a complete discussion of the			

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

235 050

next page


Y/B

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. Abstract (continued)

cont. → structure, conventions, subroutines, etc., of the LCC Model computer program. A complete listing of the FORTRAN code of the LCC Model, which contains extensive internal comments, is included in the Maintenance Manual.



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ACKNOWLEDGEMENTS

This report has been prepared by The MITRE Corporation under Project No. 6260. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Massachusetts.

The author wishes to acknowledge those who played a role in the implementation of the SEEK IGLOO Life Cycle Cost (LCC) Model as in the interactive computer program and in the production of the Maintenance Manual.

For their instructive comments on the organization of the programmed Model, I am grateful to Maj. A. E. Biery and Lt. Arnold Reyes of AFSC ESD/OCUI. For his support in the actual coding of the LCC Model, I am indebted to Harold Samson.

MITRE SEEK IGLOO Project Leader Dr. Thomas Kabaservice, Maj. A. E. Biery, Dr. Richard A. Moynihan, and John H. James reviewed the Maintenance Manual and initiated numerous improvements to the document.

Finally, I thank Joyce Hooker and Bertha Lyons for typing various sections of the document and for preparing most of the figures.

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DDC	Blk Section <input type="checkbox"/>
UNCLASSIFIED	
CLASSIFICATION	
DISTRIBUTION/AVAILABILITY CODES	
SPECIAL	
A	

78 09 11 003

PREFACE

The SEEK IGLOO Life Cycle Cost (LCC) Model is presented in three volumes, each of which serves a unique function and addresses a specific audience. Volume I, Cost Element Equations, presents a discussion of the accounting model which is used to estimate the LCC of any proposed SEEK IGLOO Radar System. This volume presents the equations for the ten Cost Elements which comprise the accounting model. Also included is a full discussion of the assumptions concerning the acquisition, operation, and logistics support of the proposed radar system which had an impact on the development of the Cost Element equations. The intended audience of Volume I is the cost analyst who is interested in a discussion of the various Cost Element equations which comprise this accounting model.

Volume II, the User's Manual, presents all the information necessary to run the computerized LCC Model effectively. Included in this volume are (1) instructions for preparing the necessary data input files, (2) an explanation of the use of the interactive capability, (3) a discussion of how to interpret the output, and (4) a complete presentation of the built-in Sensitivity Analysis capability of the LCC Model. In addition, in order to make Volume II a stand-alone document, a full discussion of the Cost Element Equations (Volume I) is included as an appendix. Thus Volume II is both a necessary and sufficient tool for utilizing the LCC Model to compute the LCC of a proposed SEEK IGLOO Radar System design.

Volume III, the Maintenance Manual, was written for the programmer who must maintain or possibly modify the FORTRAN code of the LCC Model. It contains a complete discussion of the structure, conventions, subroutines, etc. of the LCC Model computer program. A complete listing of the FORTRAN code, which contains extensive internal comments, is included in the Maintenance Manual.

TABLE OF CONTENTS

	<u>PAGE</u>
ACKNOWLEDGMENTS	1
PREFACE	2
SECTION 1. GENERAL DESCRIPTION	6
1.1 Purpose of the Maintenance Manual	6
1.2 Application of the SEEK IGLOO Life Cycle Cost Model	6
1.3 General Characteristics of the Computer Program	7
1.4 Required Equipment	7
SECTION 2. SETTING UP THE LCC MODEL FOR INTERACTIVE RUNS	8
2.1 Overview of Section II	8
2.2 General Description of the Interactive Capability	8
2.3 Setting Up an Interactive Run	8
2.4 Giving Appropriate Responses to the Program-Generated Prompts	10
SECTION 3. SETTING UP THE LCC MODEL FOR BATCH RUNS	14
3.1 Overview of Section III	14
3.2 Why Run the Program in Batch Mode?	14
3.3 Setting Up a Batch Run	14
SECTION 4. DETAILED DESCRIPTION OF THE FORTRAN CODE	17
4.1 Overview of Section IV	17
4.2 General Comments About the FORTRAN Code	17
4.3 Initialization Routines (SNSSET and INITIAL)	19
4.4 Prompting Routines (PRMPT1 to PRMPT5)	19
4.5 Read Routines (READ1 to READ5)	20
4.6 Data Input Value Error Checks (ERRCHK)	21
4.7 Auxiliary Variable Routines (AUXIL1 to AUXIL5)	21
4.8 Cost Element Routines (COST1 to COST10)	21
4.9 Non-Cost Routine (NONCOS)	21
4.10 Sensitivity Analysis Routines (DXUC to DMCI, CHCALC, TDSORT, and OSENS)	22
4.11 Printouts (ITAB1 to ITAB5 and OTAB1 to OTAB5 and OSENS)	24
4.12 Debugging Routines (APRINT and DPRINT)	24
4.13 Index Numbers	25
4.14 Common Blocks	25
4.15 Flow Chart	28

APPENDIX A: GLOSSARY

32

APPENDIX B: PROGRAM LISTING

59

LIST OF FIGURES

<u>FIGURE NUMBER</u>		<u>PAGE</u>
1	LCC Model's Configuration in Interactive Mode	9
2	Sample CLIST	10
3	Sample User Terminal Interactive Sequence	11
4	LCC Model's Configuration in Batch Mode	15
5	JCL For Batch Execution	16
6	Mini-Flow-Chart of Computer Program	18
7	Correspondence Between Common Blocks and Subroutines	27
8	Flow-Chart of Computer Program	29

SECTION I

GENERAL DESCRIPTION

1.1 Purpose of the Maintenance Manual

The purpose of this document is to enable a maintenance programmer to effectively maintain the SEEK IGLOO Interactive Life Cycle Cost (LCC) Model computer program. This maintenance manual is meant to supplement, rather than duplicate, the extensive comments included in the FORTRAN source listing. Thus, this document gives the maintenance programmer a general discussion of the computer program, allowing the programmer to refer to the internal program documentation for more detailed explanations.

Because a maintenance programmer may be required to set up the SEEK IGLOO LCC Model for future users, this document instructs the maintenance programmer on how to set up the LCC model for both interactive and batch runs (Sections 2 and 3). General and detailed comments about the FORTRAN code, a detailed flow-chart, and a cross-reference chart between common blocks and subroutines (Section 4) are included to supplement the documentation found in the FORTRAN source listing. Also included in this document are a complete glossary of FORTRAN variables (Appendix A) and a complete listing of the SEEK IGLOO Life Cycle Cost Model computer program (Appendix B).

Excluded from this document are input/output descriptions, operating instructions for users, cost element equations, an explanation of the Sensitivity Analysis capability, and a listing of error messages, which are all described in Volume II, the User's Manual.

1.2 Application of the SEEK IGLOO Life Cycle Cost Model

The SEEK IGLOO LCC Model is a tool to estimate the total Life Cycle Cost to the Government of alternative SEEK IGLOO Radar System designs. The model highlights cost elements which depend wholly or in part on the design of the radar and those which depend on Contractor-provided data which affect support concepts. Thus, the LCC Model will be useful to both the Government and the Contractor in estimating the Life Cycle Cost of potential radar system designs.

The SEEK IGLOO LCC Model computer program consists of two main components. The first component implements an accounting model which computes the Life Cycle Cost of the SEEK IGLOO radar system based on the values of the input parameters corresponding to a particular system design. The output of this first component gives the total LCC and also the LCC broken out into ten component Cost Elements, into Development, Production, and Support Costs, and into cost incurred at different locations (e.g.,

radar sites versus repair facilities). The breakdown of Life Cycle Cost into various categories should help the user identify areas in which design and cost trade-offs should be considered.

The second component of the LCC computer program provides Sensitivity Analysis computations to be used as an additional aid in trade-off considerations. More specifically, for certain selected contractor parameter inputs (e.g., UC(i) - the unit cost of the ith assembly, or FR(i) - the failure rate of the ith assembly) this program component computes the average change (either positive or negative) in total LCC which is produced by a fractional increase in the value of the given parameter.

1.3 General Characteristics of the Computer Program

The program is highly modular and has a top-down structure. The code was written in ANS FORTRAN (X3.9-1966), except for its use of the NAMELIST feature, so that it would be compatible with almost any FORTRAN compiler. The code has a great deal of internal documentation and has consistent naming conventions. These general aspects of the computer program are detailed in Section 4.2.

1.4 Required Equipment

To run the program in the interactive environment for which it was designed, the user will need a computer facility with the capability of running a FORTRAN program on-line (i.e., in interactive mode) with 256K bytes of memory, six input files, and two output files. Five of the input files are assigned to card-image data sets. The other (sixth) input file and one of the two output files are assigned to the user's terminal, which must have at least 72 columns. The remaining output file is assigned to an off-line printer of at least 121 character width, including a carriage control character.

If the user chooses to run the program in a batch environment, then the user's computer facility will need the capability of running a batch FORTRAN program with 256K bytes of memory, six input files, and two output files. All six input files are assigned to card-image data sets. One output file can be assigned to a data set, to a dummy file, or to a printing device. The other output file is assigned to an off-line printer of at least 121 character width, including a carriage-control character.

SECTION II

SETTING UP THE LCC MODEL FOR INTERACTIVE RUNS

2.1 Overview of Section II

The purpose of this section is to enable the maintenance programmer to set up the SEEK IGLOO LCC Model for interactive use on an IBM 370/158 computer. Section 2.2 includes a very general description of the interactive capability (a detailed description is found in the User's Manual). Section 2.3 lists the file allocations necessary for an interactive run and includes a sample IBM exec file (called a CLIST under IBM's Time Sharing Option). Finally, Section 2.4 illustrates the inputs and outputs for a sample terminal session.

2.2 General Description of the Interactive Capability

Figure 1 shows the LCC Model's configuration when used in interactive mode (the word "CHANNEL" in Figure 1 refers to the data set reference number found in the FORTRAN code). The interactive capability allows for the repetitive on-line (i.e., interactive) execution of the LCC model. The model reads five "standard" data files which contain a complete set of inputs (these five files are: Miscellaneous Scalar Inputs, Support Equipment Inputs, Functional Area Inputs, LRU Equipment Inputs, and LRU Maintenance Inputs). After reading the five files, the program allows the user to override any of these "standard" inputs by entering new values from his terminal. The program will then use these interactive inputs to calculate and print the LCC. Thus, the interactive feature allows the user to change his inputs without altering his five "standard" input files.

2.3 Setting Up an Interactive Run

After having set up the five "standard" data files (see Section 3.2 of the User's Manual), to run the SEEK IGLOO LCC Model in an interactive mode, the user must perform (or have performed for him or her) the following steps:

1. Log his or her terminal onto the computer with 256K bytes of memory.
2. Assign FORTRAN input/output channel 11 to the Miscellaneous Scalar Inputs data set.
3. Assign FORTRAN input/output channel 12 to the Support Equipment Inputs data set.
4. Assign FORTRAN input/output channel 13 to the Functional Area Inputs data set.

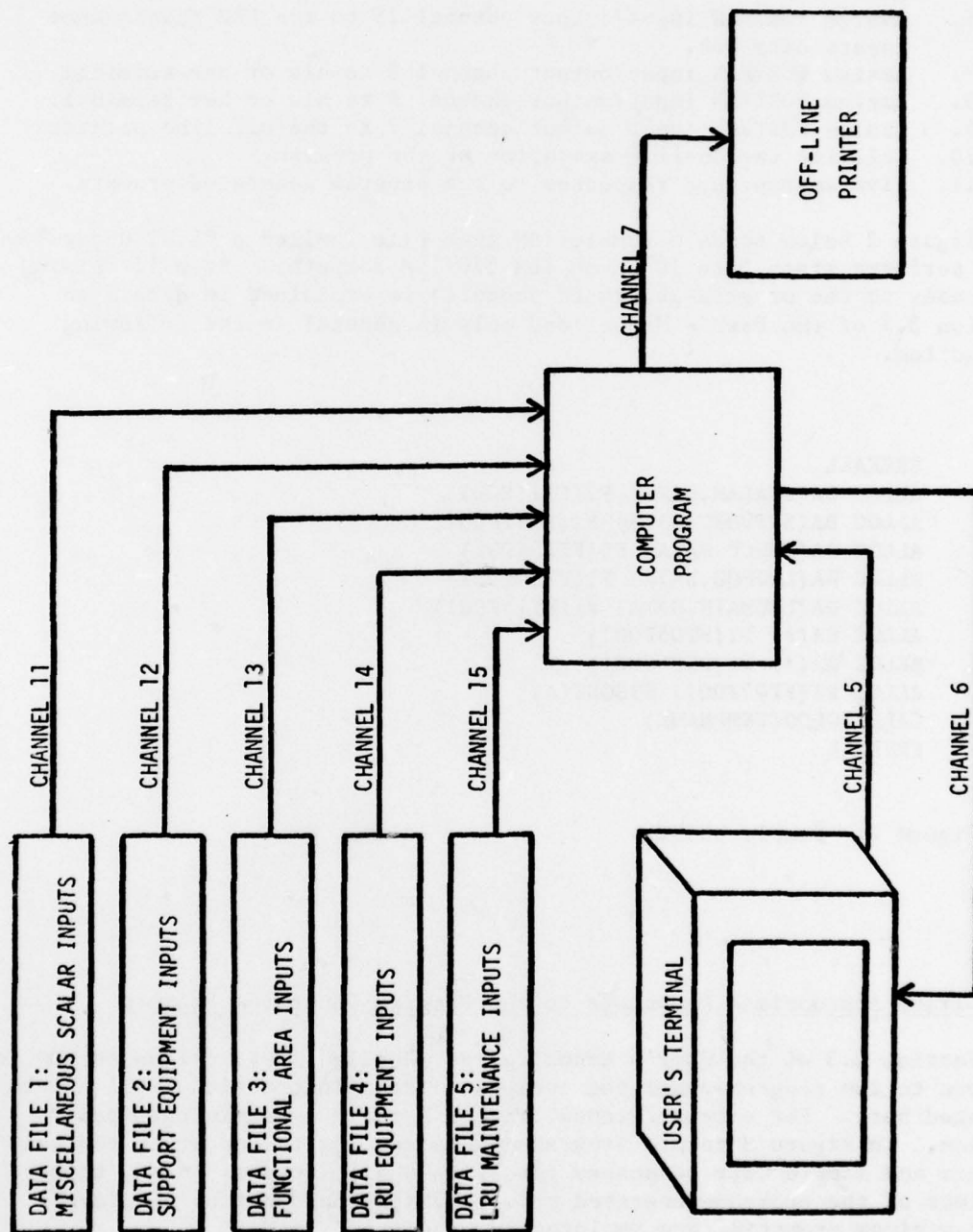


Figure 1: LCC Model's Configuration in Interactive Mode

5. Assign FORTRAN input/output channel 14 to the LRU Equipment Inputs data set.
6. Assign FORTRAN input/output channel 15 to the LRU Maintenance Inputs data set.
7. Assign FORTRAN input/output channel 5 to his or her terminal.
8. Assign FORTRAN input/output channel 6 to his or her terminal.
9. Assign FORTRAN input/output channel 7 to the off-line printer.
10. Call for the on-line execution of the program.
11. Give appropriate responses to the program generated prompts.

Figure 2 below shows a sample IBM exec file (called a CLIST under TSO) that performs steps 2 to 10 on an IBM 370/158 computer. Step 11 (giving responses to the program-generated prompts) is explained in detail in Section 3.3 of the User's Manual and only in general in the following subsection.

```

FREEALL
ALLOC DA(SCALAR.DATA) FI(FT11F001)
ALLOC DA(SUPPORT.DATA) FI(FT12F001)
ALLOC DA(FUNCT.DATA) FI(FT13F001)
ALLOC DA(LRUEQU.DATA) FI(FT14F001)
ALLOC DA(LRUMAIN.DATA) FI(FT15F001)
ALLOC DA(*) FI(FT05F001)
ALLOC DA(*) FI(FT06F001)
ALLOC FI(FT07F001) SYSOUT(A)
CALL IGLOO(TEMPNAME)
FREEALL

```

Figure 2: Sample CLIST

2.4 Giving Appropriate Responses to the Program-Generated Prompts

Section 3.3 of the User's Manual gives detailed instructions on how to respond to the program-generated prompts. These instructions will not be repeated here. For easy reference, Figure 3 shows a sample terminal session. In Figure 3 sample program-generated prompts appear in capital letters and sample user responses appear in small letters. Also, those portions of the program-generated prompts which would not be displayed under minimum prompting are enclosed in brackets.

SEEK IGLOO LIFE CYCLE COST MODEL

MINIMUM OR MAXIMUM PROMPTING (MIN OR MAX)-?

max

OUTPUT AT TERMINAL (Y OR N)-?

y

OUTPUT ON OFF-LINE PRINTER (Y OR N)-?

y

SUBMIT A TITLE FOR THIS RUN:

first run

SET EXIT=1 IN EITHER NAMELIST IF YOU WANT TO EXIT.

NAMELIST /GO/ CONTAINS ALL VARIABLES FOUND IN THE INPUT FILES. AT THIS POINT, NAMELIST /GO/ VARIABLES CONTAIN VALUES AS IN THE INPUT FILES. TO USE THESE VALUES, SUBMIT AN EMPTY NAMELIST /GO/. TO OVERRIDE ANY OF THESE VALUES, SUBMIT A NON-EMPTY NAMELIST /GO/.

SUBMIT NAMELIST /GO/ IN NAMELIST FORMAT:

\$go uc(2)=275, fac(1)=3000, qpa(5)=0 \$end

NAMELIST /SENS/ CONTAINS VARIABLES THAT CONTROL THE DISPLAY OF THE SENSITIVITY ANALYSIS. AT THIS POINT, THE TERMINAL DISPLAYS SENSITIVITY WITH RESPECT TO ONLY GLOBAL FAILURE RATE (K) AND GLOBAL EQUIPMENT COST (XUC FACTOR). FOR THE SAME TERMINAL DISPLAY, SUBMIT AN EMPTY NAMELIST /SENS/. FOR A DIFFERENT TERMINAL DISPLAY, SUBMIT A NON-EMPTY NAMELIST /SENS/.

Figure 3: Sample User Terminal Interactive Sequence

SUBMIT NAMELIST /SENS/ IN NAMELIST FORMAT:

\$sens ldfr=6, ldmci=6, ldsrts=6 \$end

LCC COMPLETED.

IF YOU WISH TO EXIT, HIT -E-, THEN HIT -RETURN-; OTHERWISE,
ADJUST TERMINAL TO NEW PAGE AND HIT -RETURN-.

(LCC OUTPUT TABLE)

IF YOU WISH TO EXIT, HIT -E-, THEN HIT -RETURN-; OTHERWISE,
ADJUST TERMINAL TO NEW PAGE AND HIT -RETURN-.

(SENSITIVITY ANALYSIS TABLE)

ANOTHER RUN (Y OR N)-?

y

AT THIS POINT, VARIABLE VALUES ARE AS THEY WERE AFTER THE LAST
NAMELISTS WERE SUBMITTED. DO YOU WISH TO RESET NAMELIST /GO/
VARIABLES TO THE VALUES FOUND IN THE INPUT FILES (Y OR N)-?

n

MINIMUM OR MAXIMUM PROMPTING (MIN OR MAX)-?

max

OUTPUT AT TERMINAL (Y OR N)-?

y

OUTPUT ON OFF-LINE PRINTER (Y OR N)-?

y

Figure 3: Sample User Terminal Interactive Sequence (Continued)

SUBMIT A TITLE FOR THIS RUN:

second run

SET EXIT=1 IN EITHER NAMELIST IF YOU WANT TO EXIT.

NAMelist /GO/ CONTAINS ALL VARIABLES FOUND IN THE INPUT FILES. AT THIS POINT, NAMELIST /GO/ VARIABLES ARE AS THEY WERE AFTER THE LAST NAMELIST /GO/ WAS SUBMITTED. TO USE THESE VALUES, SUBMIT AN EMPTY NAMELIST /GO/. TO OVERRIDE ANY OF THESE VALUES, SUBMIT A NON-EMPTY NAMELIST /GO/.

SUBMIT NAMELIST /GO/ IN NAMELIST FORMAT:

\$go \$end

NAMelist /SENS/ CONTAINS VARIABLES THAT CONTROL THE DISPLAY OF THE SENSITIVITY ANALYSIS. AT THIS POINT, THE TERMINAL SENSITIVITY DISPLAY IS AS IT WAS ON THE PREVIOUS RUN. FOR THE SAME TERMINAL DISPLAY, SUBMIT AN EMPTY NAMELIST /SENS/. FOR A DIFFERENT TERMINAL DISPLAY, SUBMIT A NON-EMPTY NAMELIST /SENS/.

SUBMIT NAMELIST /SENS/ IN NAMELIST FORMAT:

\$sens finc=.20, ldfr=0 \$end

LCC COMPLETED.

IF YOU WISH TO EXIT, HIT -E-, THEN HIT -RETURN-; OTHERWISE, ADJUST TERMINAL TO NEW PAGE AND HIT -RETURN-.

(LCC OUTPUT TABLE)

IF YOU WISH TO EXIT, HIT -E-, THEN HIT -RETURN-; OTHERWISE, ADJUST TERMINAL TO NEW PAGE AND HIT -RETURN-.

(SENSITIVITY ANALYSIS TABLE)

ANOTHER RUN (Y OR N)-?

n

Figure 3: Sample User Terminal Interactive Sequence (Concluded)

SECTION III

SETTING UP THE LCC MODEL FOR BATCH RUNS

3.1 Overview of Section III

The purpose of this section is to instruct the maintenance programmer on how to set up the SEEK IGLOO LCC Model for batch runs. Section 3.2 explains why a user might want to run the LCC Model in a batch environment. Section 3.3 explains the file allocations required for a batch run and includes sample Job Control Language for running the program on an IBM 370/158 computer.

3.2 Why Run the Program in Batch Mode?

Batch execution of the LCC Model allows the user to run the LCC Model without having to spend a great deal of time at his terminal. Two types of users will want to run the LCC Model in batch mode: the user whose computer facility cannot run the program interactively; and the user who prefers the slower, more methodical, and usually less expensive characteristics of batch execution.

3.3 Setting Up a Batch Run

Appendix E of the SEEK IGLOO LCC Model User's Manual describes in detail the content of the data files needed for a batch run. This subsection supplies a general description of the content of the data files needed for a batch run, a description of the output files from a batch run, and a description of the JCL need for a batch run.

Figure 4 shows the LCC Model's configuration in batch mode (the word "CHANNEL" in Figure 4 again refers to the data set reference number found in the FORTRAN code). The content of the five "standard" data files (Miscellaneous Scalar Inputs, Inputs by Support Equipment type, Inputs by Functional Area, LRU Equipment Inputs, and LRU Maintenance Inputs) is the same for a batch run as for an interactive run. A sixth input data file needs to be created for a batch run, however. It contains a listing of the responses to program-generated prompts which the user would have given had the user run the program interactively.

As with an interactive run, there are two output files from a batch run. One of the files (channel 7) should be assigned to the off-line printer. The other file (channel 6) can be assigned to one of three things: to a dummy file (in case the user requests off-line printing only); to a permanent-resident data set (if, for some reason, the user wishes to save the prompts and output that would have gone to the terminal

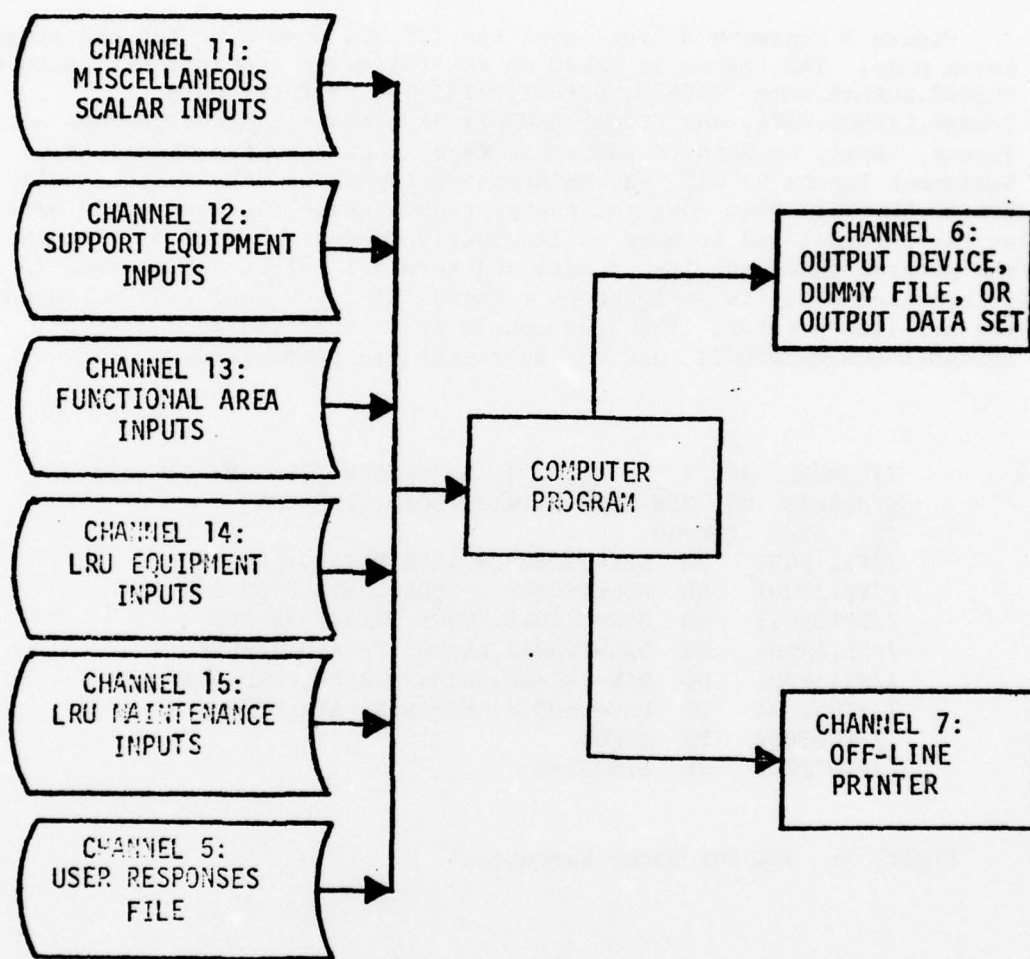


Figure 4: LCC Model's Configuration in Batch Mode

in interactive mode); or to a printing device (if the user wishes hard-copy of the prompts and output that would have gone to the terminal in interactive mode).

Figure 5 contains a listing of the IBM JCL needed to run the program in batch mode. The figure is based on the following assumptions: data sets TS0848.SCALAR.DATA, TS0848.SUPPORT.DATA, TS0848.FUNCT.DATA, TS0848.LRUEQU.DATA, and TS0848.LRUMAIN.DATA contain Miscellaneous Scalar Inputs, Inputs by Support Equipment type, Inputs by Functional Area, Equipment Inputs by LRU, and Maintenance Inputs by LRU, respectively; TS0848.RESPONSE.DATA contains the responses which the user would have typed at the terminal had he been in interactive mode; channel 6, through which the program sends out its prompts and terminal output tables when in interactive mode, is assigned to a dummy file; channel 7 is assigned to the off-line printer; the load module to be executed is TS0848.IGLOO.LOAD(GO); and all data sets are catalogued.

```
//SAMPLE JOB (_____,_____,_____), 'H. SAMSON', REGION=256K, etc...
//JOB LIB DD DSN=TS0848.IGLOO.LOAD, DISP=SHR
// EXEC PGM=GO
//FT11F001 DD DSN=TS0848.SCALAR.DATA, DISP=SHR
//FT12F001 DD DSN=TS0848.SUPPORT.DATA, DISP=SHR
//FT13F001 DD DSN=TS0848.FUNCT.DATA, DISP=SHR
//FT14F001 DD DSN=TS0848.LRUEQU.DATA, DISP=SHR
//FT15F001 DD DSN=TS0848.LRUMAIN.DATA, DISP=SHR
//FT05F001 DD DSN=TS0848.RESPONSE.DATA, DISP=SHR
//FT06F001 DD DUMMY
//FT07F001 DD SYSOUT=A
```

Figure 5: JCL for Batch Execution

SECTION IV

DETAILED DESCRIPTION OF THE FORTRAN CODE

4.1 Overview of Section IV

The various subsections of Section IV each elaborate on a single aspect of the FORTRAN code. The discussions are meant to supplement, rather than duplicate, the extensive documentation found within the FORTRAN code. Section 4.2 comments on the code in general. Sections 4.3 to 4.12 each elaborate on a single subroutine or a related group of subroutines. Section 4.13 is concerned with the program's use of index numbers. Section 4.14 discusses the program's use of common blocks and includes a chart showing the correspondence between subroutines and common blocks. Finally, Section 4.15 contains a flow-chart of the program and a discussion of the flow-chart.

4.2 General Comments About the FORTRAN Code

The following general comments about the FORTRAN code might be helpful to the maintenance programmer.

1. The program was written in ANS (American National Standard: X3.9-1966) FORTRAN so that it would be compatible with as many machines as possible. The following characteristics exemplify the attempted compatibility:
 - a. All mixed-mode arithmetic has explicit integer-to-real conversions.
 - b. All variables take up a single word of memory (no Integer*2 or Real*8 variables).
 - c. Characters are stored one character per memory location (hence, a 28-character string is stored in a 28-element array).
2. The code has a top-down structure that is depicted in the mini-flow chart of Figure 6. One pass through the loop in Figure 6 represents a single calculation of the LCC. The user can make multiple passes through the loop (thereby generating multiple calculations of the LCC) by successively responding 'y' to the prompt 'ANOTHER RUN (Y OR N)-?'.

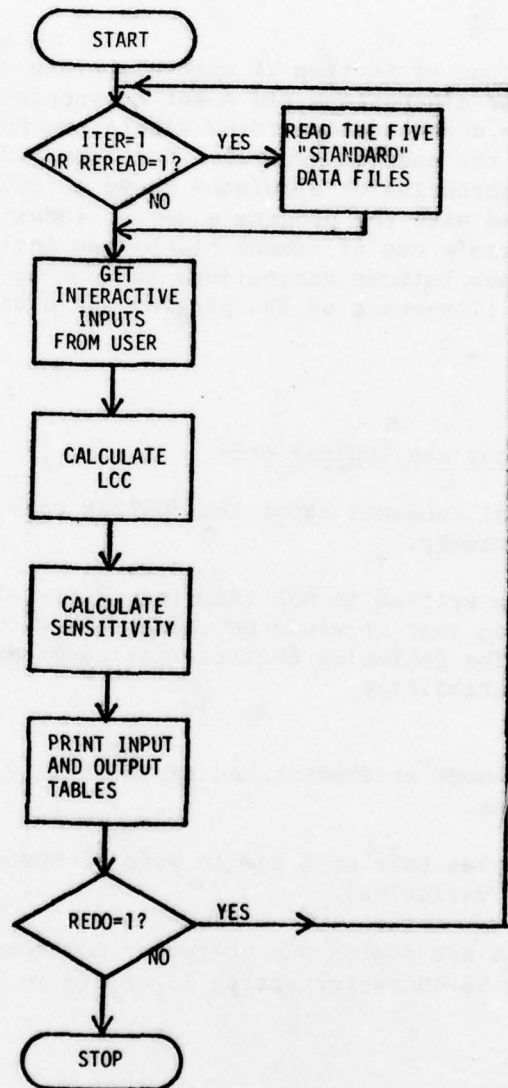


Figure 6: Mini-Flow Chart of LCC Model

3. The program is highly modular. The MAIN routine consists almost entirely of subroutine calls, with each subroutine performing a single function. For instance, for each of the five data files that must be read, there is a single subroutine which does nothing but read the given data file and check for index number errors. Similarly, for each cost element there is a single subroutine that does nothing but calculate the given cost element.
4. The program has extensive in-line comments. The maintenance programmer should have little difficulty determining the intent of any particular block of code.

4.3 Initialization Routines (SNSSET and INITAL)

Subroutine SNSSET initializes the parameters which control the Sensitivity Analysis printout (refer to subsection 4.10 for a discussion of the Sensitivity Analysis). LDERV is set to 12; FINC is set to .1; all other control parameters (LDFR, LDUC, LDWOR, LDFPR, LDCMH, LDDMH, LDRM, LDSRTS, LDCRTS, LDDRTS, LDMCI) are set to zero. These initial values cause:

1. the "default" off-line Sensitivity Analysis printout to show the 12 most significant sensitivity calculations for each LRU-indexed parameter which is subject to Sensitivity Analysis.
2. the "default" user terminal Sensitivity Analysis printout to show Sensitivity calculations only with respect to the K-factor and the XUC-factor.

For off-line output and for terminal output, the "default" printout is displayed unless the user gives specific interactive instructions through NAMELIST /SENS/ (Namelist /SENS/ contains parameters that control the Sensitivity Analysis printout).

Certain other non-sensitivity analysis variables are given initial values in subroutine INITAL. The variables which are initialized in subroutine INITAL include MAXHRS, MAXFA, MAXLR, MAXSE, CONFLO, FAC(.), NSES(j), NSEC(j), NSED(j), UC(i), FPR(i), CMH(i), DMH(i), RM(i), MCI(i), QPA(i), and MTBI(i).

4.4 Prompting Routines (PRMPT1 to PRMPT5)

PRMPT1 to PRMPT5 are subroutines that send prompts to the user and receive the user's responses.

PRMPT1 sets variables MAXPMT, PRNT, and REREAD. MAXPMT is set to 1 if the user responds 'min' to the prompt: "MINIMUM or MAXIMUM PROMPTING (MIN OR MAX) -?". PRNT is set according to the user's responses to the prompt: "OUTPUT AT TERMINAL (Y OR N) -?" and "OUTPUT ON OFFLINE PRINTER(Y OR N) -?". It receives the value 0 if the user requested terminal output only; 1 for off-line output only; and 2 for both terminal and off-line output. If the user answered 'n' to both questions, PRNT=0. REREAD is set to 1 on the second and subsequent runs if the user responds 'y' to the prompt: "DO YOU WISH TO RESET NAMELIST /GO/ VARIABLES TO THE VALUES FOUND IN THE INPUT FILES (Y OR N) -?". REREAD is not relevant on the first run of the LCC.

PRMPT2 asks the user to submit NAMELISTs /GO/ and /SENS/. NAMELIST /GO/ contains LCC inputs. The values which are inputted through NAMELIST /GO/ override the values found in the five "standard" data files (or the values used in the previous pass if this pass is not the first and if the user did not choose to reread the five "standard" data files at the start of this pass). Namelist /SENS/ contains parameters that control the Sensitivity Analysis printout.

PRMPT3 tells the user that the LCC has been completed and allows the user to adjust the terminal to a new page before Output Table 1 is printed. PRMPT4 is called if the user requested terminal printing. It allows the user to adjust the terminal to a new page before the Sensitivity Analysis printout. PRMPT5 asks the user if another run is desired.

The user can exit from the program in PRMPT2 by setting EXIT=1 in either namelist. The user can exit from PRMPT3 or PRMPT4 by hitting "e" and then carriage return in response to the program-generated message. These escapes send the program to PRMPT5.

Maximum prompting gives the user a longer, more descriptive set of program-generated prompts. Max-prompting can be switched to min-prompting on subsequent passes within a single interactive session, whereas a switch from min-prompting to max-prompting is not possible.

4.5 Read Routines (READ1 to READ5)

Each of the READ subroutines reads a single input file and checks for index number errors. The correspondence is as follows:

READ1 reads UNIT 11: Miscellaneous Scalar Inputs
READ2 reads UNIT 12: Support Equipment Inputs
READ3 reads UNIT 13: Functional Area Inputs
READ4 reads UNIT 14: LRU Equipment Inputs
READ5 reads UNIT 15: LRU Maintenance Inputs

The proper formatting of data within each of the above input files is described in detail in Section 3.2 of the User's Manual. In particular,

the data inputs in UNITS 12 through 15 are ordered on the basis of their index numbers. The rules governing the use of index numbers on these data input files are explained in Section 3.2 of the User's Manual and in Section 4.13 of this document.

4.6 Data Input Value Error Checks (ERRCHK)

Additional error checks involving the values of data parameters are performed by subroutine ERRCHK. Two types of errors are possible: warning and fatal. If an error is found, the program generates a message and increments the appropriate internal counter (IWARN for a warning error; IERROR for a fatal error). A warning error results in a warning message and in the assigning of a default value to the error-generating parameter; a fatal error results in an error message and no calculation of the LCC (the program skips the LCC calculation by jumping to PRMPT5). Section 3.3.1 of the User's Manual contains a list of the data input value error messages.

4.7 Auxiliary Variable Routines (AUXIL1 to AUXIL5)

All auxiliary variables are calculated according to the equations found in Appendix B of the SEEK IGL00 User's Manual. The only complex routines involve the calculations of Z, STK1, and STK, which are done in subroutine AUXIL5.

AUXIL5 calculates Z from the value the user gives to confidence factor CONF. Z is the approximate upper limit on the integral of the standard normal distribution such that the value of the definite integral from negative infinity to Z equals CONF. The integral is approximated using the trapezoidal rule.

The stocking factors (STK1 and STK) are calculated according to the equations in Section C.4 of the User's Manual. The User's Manual assumes convergence of the series 1.1 in Section C.4. However, if the series doesn't converge after 200 iterations, the program generates an error message.

4.8 Cost Element Routines (COST1 to COST10)

Subroutines COST1 to COST10 each calculate a single cost element of the LCC. The program calculates these cost elements according to their equations as given in Appendix C of the User's Manual.

4.9 Non-Cost Routine (NONCOS)

NONCOS calculates certain variables not used in the LCC calculations. These variables have to do with maintenance man-hours and, although they are not of interest in terms of dollars, they are still of interest in determining whether certain maintenance constraints have been met. Variables CLH, PMFA(j), SMMH, SPMH, and STMH are calculated in NONCOS. CLH, SMMH, SPMH, and STMH are checked against certain maintenance man-hour constraints in subroutine OTAB1. If they fail the required test, OTAB1 generates an error message. PMFA(j) is the annual preventive maintenance man-hours expended at each radar site within Functional Area j. It is one of the outputs in Output Table 4.

4.10 Sensitivity Analysis Routines (DXUC to DMCI, CHCALC, TDSORT, and OSENS)

Subroutines DXUC, DFR, DK, DUC, ..., DMCI each calculate the Sensitivity of the LCC with respect to a single parameter (XUC, FR(i), K, UC(i), ..., MCI(i), respectively). Subroutine OSENS prints the Sensitivity Analysis results. Except for DXUC and DK, these calculations are indexed by LRU type.

The Sensitivity of LCC with respect to a given parameter xxx is the average change in LCC which results from a given change in the value of parameter xxx. In effect, these calculations are equivalent to computing derivatives. Hence, in the internal program documentation, the sensitivity (or change) in LCC with respect to a given parameter xxx is referred to as "the derivative of LCC with respect to parameter xxx." In particular, the sensitivity of LCC with respect to a given parameter xxx is assigned to the variable TDxxx. Also, the value of the variable FINC equals the fractional increase in each parameter which is used to calculate each resulting change in LCC.

For example, if FINC=.1, then in the sensitivity calculation with respect to the parameter FPR for LRU type 10, the value of TDFPR(10) equals the approximate change in LCC that would result from a 10% increase in the false-pull-rate of LRU type 10.

The Sensitivity Analysis calculations are the most complex part of the program. The following comments supplement the Sensitivity Analysis description that is found in Appendix D of the User's Manual.

1. Naming Conventions - Naming conventions for the Sensitivity Analysis are as follows: the subroutines that calculate Sensitivity with respect to xxx have the form Dxxx (such as DXUC, DFR, DK, ..., DMCI); the variables (usually arrays) which hold the actual value of the derivative (i.e., sensitivity) have the form TDxxx (such as TDXUC, TDFR(i), TDK, ..., TDMCI(i)); the variable which tells the number of "most significant" derivatives of each type that the user wishes to be printed on the off-line printer is called LDERV; the variables which tell the number of

"most significant" derivatives of type xxx which the user wishes to see at the terminal (or on the off-line output if it exceeds LDERV) have the form LDxxx (such as LDFR, LDUC, ..., LDMCI); the variables which hold the sorted index numbers (the index number of the most significant derivative to the index number of the least significant derivative) have the form IDxxx (such as IDFR(i), IDUC(i), ..., IDMCI(i)); and, for certain parameters (WOR(i), SRTS(i), CRTS(i), DRTS(i), MCI(i)), the variables which hold the change in the parameter to which the projected LCC change corresponds have the form WF(i), SFSRTS(i), CFCRTS(i), DFDRTS(i), MCIC(i), respectively.

2. Sensitivity Analysis Printout - LDERV and the LDxxx variables control the printout of the Sensitivity Analysis calculations with respect to all parameters which are indexed by LRU type (only sensitivity with respect to K-factor and XUC-factor are not indexed by LRU type). These variables are initialized in subroutine SNSSET and are updated by user inputs through NAMELIST /SENS/ in subroutine PRMPT2. Sensitivity output at the terminal consists of global sensitivity with respect to both the K-factor and XUC-factor plus whatever additional sensitivity calculations which the user requests through NAMELIST /SENS/. The additional output depends on the values of the LDxxx variables (part 1 above describes the significance of the LDxxx variables). Sensitivity output on the off-line printer consists of global sensitivity with respect to the XUC-factor and the K-factor plus the LDERV "most significant" derivatives with respect to each of the other parameters (unless any particular LDxxx values exceed LDERV, in which case LDxxx derivatives would be printed off-line for factor xxx).
3. Skipping Sensitivity Analysis Subroutines - In general, if no sensitivity output is requested for a particular sensitivity parameter type, then the corresponding subroutine is not called. Exceptions are subroutines DXUC, DK, and DFR. Subroutines DXUC and DK are always called because TDXUC and TDK are always printed. Subroutine DFR is always called because TDFR(i) is used to calculate TDK.
4. Number of Sensitivity Outputs Per Line of Output - For any particular LRU-indexed parameter, 6 sensitivity values are printed per line of output at the terminal; 12 values per line appear in the off-line printout.
5. Subroutine CHCALC - Subroutines DWOR, DSRTS, DCRTS, and DDRTS call subroutine CHCALC to calculate the changes in LCC with respect to the parameters WOR(i), SRTS(i), CRTS(i), and DRTS(i), respectively. For each LRU type, the calling routine assigns values to variables CF, DF, SF, CCF, DCF, and SCF, which represent the changes in the internally calculated (in subroutine AUXIL1)

data parameters CRTS(i), DRTS(i), SRTS(i), CCOND(i), DCOND(i), and COND(i), respectively. Subroutine CHCALC then uses these variables as parameters to calculate CHLCC, which is returned to the calling routine to be used as the computed change in LCC. Variables CF, DF, SF, CCF, DCF, SCF and CHLCC are passed through common block CHLCC.

6. Subroutine TDSORT - For those subroutines which calculate sensitivity indexed by LRU type, the subroutine (Dxxx) first calculates sensitivity of LCC with respect to factor xxx for all LRU types. Then the subroutine calls subroutine TDSORT, which bubbles up to the top of array IDxxx the index numbers of the LRU types for which LCC is most sensitive for factor xxx. To save on execution time, only LLL index numbers are actually bubbled up to the top of array IDxxx, where $LLL = \max(\# \text{ of off-line outputs for factor xxx}, \# \text{ of terminal outputs for factor xxx})$.

4.11 Printouts (ITAB1 to ITAB5 and OTAB1 to OTAB5 and OSENS).

Each subroutine in this group produces a single table of the LCC Model output (see Appendix F of the User's Manual for a sample off-line output listing). Terminal output consists only of Output Table 1 (OTAB1) and the Sensitivity Analysis Table (OSENS). Off-line output consists of Input Tables 1 to 5 (ITAB1 to ITAB5), Output Tables 1 to 5 (OTAB1 to OTAB5), and the Sensitivity Analysis Table (OSENS). The input tables print the data that was used to calculate the LCC. The output tables show the result of the LCC Model's calculations.

Besides these tables, certain other outputs may be sent to either the terminal or off-line printer. Input error messages are printed at the terminal (always) and on the off-line printer (if off-line output was chosen). Input error messages are generated by the read subroutines (READ1 to READ5) and by subroutine ERRCHK. They are printed before the input and output tables. Maintenance man-hour constraint error messages accompany the LCC output whenever the constraints are not met. Maintenance man-hour error messages are generated by OTAB1.

4.12 Debugging Routines (APRINT and DPRINT)

Two FORTRAN debugging subroutines can be included in the program by adding two call statements to the program (CALL APRINT and CALL DPRINT) after the calls of the Sensitivity Analysis subroutines. APRINT prints the values of all auxiliary variables and cost elements. DPRINT prints the values of all Sensitivity Analysis variables.

4.13 Index Numbers

The following comments concern index numbers used with the five standard data input files (see Section 4.5 above):

1. The Unit 13 index numbers (Functional Area Inputs) must be consecutive starting with 1.
2. The Unit 12, 14, and 15 index numbers (Support Equipment Inputs, LRU Equipment Inputs, and LRU Maintenance Inputs) must be increasing, although gaps are allowed. The largest index number cannot exceed a pre-assigned maximum value (150, 200, and 200, respectively).
3. The Unit 14 file (LRU Equipment Inputs) and the Unit 15 file (LRU Maintenance Inputs) must have identical sequences of index numbers.
4. Calculations by LRU and Support Equipment type have loops from 1 to the largest respective index number. Missing items are skipped due to zero values in the quantity arrays (QPA(i) for LRUs; NSEC(), NSES(), and NSED() for Support Equipment).

4.14 Common Blocks

The following common block conventions were used when programming the LCC Model:

1. A single common block holds all data read from a particular READx subroutine (common blocks RD1 to RD5 correspond to subroutines READ1 to READ5).
2. A single common block holds all of the auxiliary variables that were calculated in a single auxiliary variables subroutine (common blocks AUX1 to AUX5 correspond to subroutines AUXIL1 to AUXIL5).
3. A single common block holds all cost element summation variables for a single cost element (common blocks C1 to C10 correspond to subroutines COST1 to COST10).
4. A single common block holds all variables having to do with a particular Sensitivity Analysis subroutine (common blocks TDXUC to TDMCI correspond to subroutines DXUC to DMCI).
5. The other common blocks escape convention: CNTL holds various program-flow parameters; TITL holds the identifying title of a particular LCC pass; ERROR holds the variables that count the

number of error messages generated; INIT holds miscellaneous internally set variables which never change value; SENS holds the Sensitivity Analysis control parameters; and CHLCC holds parameters which are passed to and from subroutine CHCALC.

The dimensioning of the elements of the various common blocks is worth noting. All variables which are indexed by LRU are dimensioned to 200 (common blocks which contain elements which are dimensioned by LRU include RD4, RD5, AUX1-5, C2, C3, C4, C9, and TDFR-TDMCI); all variables which are indexed by support equipment type are dimensioned to 150 (common blocks which contain arrays that are dimensioned by support equipment type include RD2); all variables which are indexed by functional area type are dimensioned up to 10 (common blocks which contain arrays which are dimensioned by functional area type include RD3 and NONCOS). Figure 7 contains a cross-reference chart showing the correspondence between subroutines and common blocks.

SUBROUTINEMAIN

SNSSET
PRMPT1
PRMPT2
PRMPT3
PRMPT4
PRMPT5
TITLE
INITAL
READ1
READ2
READ3
READ4
READ5
ITAB1
ITAB2
ITAB3
ITAB4
ITAB5
ERRCHK
AUXIL1
AUXIL2
AUXIL3
AUXIL4
AUXIL5
COST1
COST2
COST3
COST4
COST5
COST6
COST7
COST8
COST9
COST10
NONCOS
DXUC
DFR

DK
DUC
DWOR
DFPR

DCMH
DDMH
DRM
DSRTS
DCRTS

COMMON BLOCKSALL COMMON BLOCKS

SENS
CNTL,TITL
CNTL,RD1,RD2,RD3,RD4,RD5,SENS
CNTL
CNTL
CNTL
TITL
INIT,RD1,RD2,RD4,RD5
RD1
CNTL,ERROR,INIT,RD2
CNTL,ERROR,INIT,RD2
CNTL,ERROR,INIT,RD4
CNTL,ERROR,INIT,RD4,RD5
RD1
RD2
RD3
RD4
INIT,RD4,RD5
CNTL,ERROR,INIT,RD1,RD4,RD5
RD4,RD5,AUX1
RD1,RD4,RD5,AUX2
RD4,RD5,AUX1,AUX2,AUX3
RD1,RD4,RD5,AUX1,AUX2,AUX4
CNTL,INIT,RD1,RD4,RD5,AUX2,AUX4,AUX5
RD1,C1
RD1,RD4,AUX2,AUX3,AUX5,C2
RD1,RD4,AUX4,AUX5,C3
RD1,RD4,AUX1,AUX2,C4
RD1,C5
RD1,RD2,C6
RD1,C7
RD1,C8
RD1,RD4,AUX1,C9
RD1,RD3,C10
RD1,RD3,RD4,RD5,AUX1,AUX2,AUX3,AUX5,NCOS
RD1,SENS,C3,C4,TDXUC
CNTL,RD1,RD4,RD5,SENS,AUX1,AUX2,AUX3,AUX4,
AUX5,TDFR
RD4,TDFR,TDK
CNTL,RD1,RD4,SENS,AUX1,AUX2,AUX4,AUX5,TDUC
CNTL,RD4,RD5,SENS,AUX1,CHLCC,TDWOR
CNTL,RD1,RD4,RD5,SENS,AUX1,AUX2,AUX4,AUX5,
TDFPR
CNTL,RD1,RD4,RD5,SENS,AUX1,AUX2,TCMH
CNTL,RD1,RD4,RD5,SENS,AUX1,AUX2,TDDMH
CNTL,RD1,RD4,SENS,AUX1,AUX2,TDRM
CNTL,RD4,SENS,AUX1,CHLCC,TDSRTS
CNTL,RD4,SENS,AUX1,CHLCC,TD CRTS

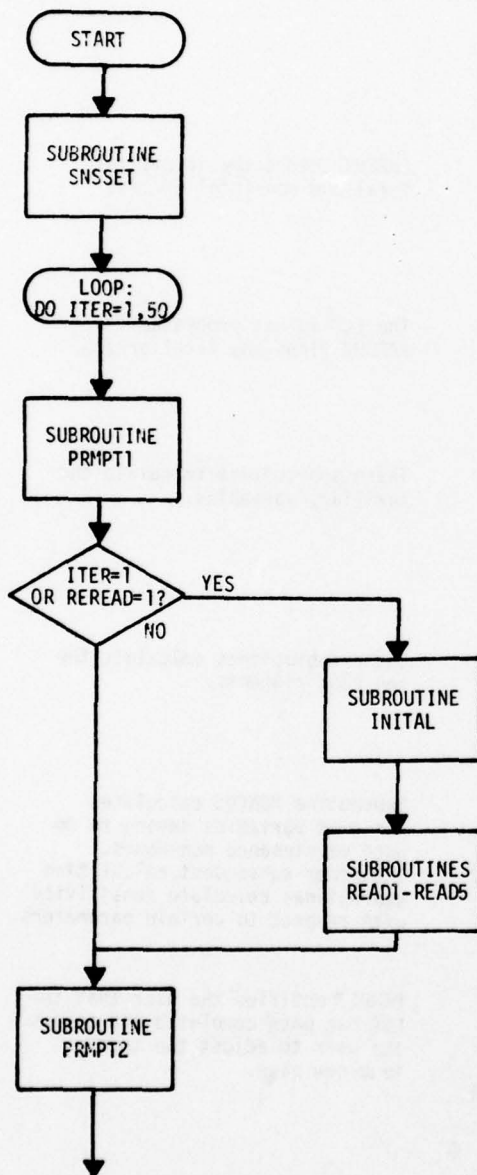
Figure 7: Correspondence Between Common Blocks and Subroutines

DDRTS	CNTL, RD4, SENS, AUX1, CHLCC, TDDRTS
DMCI	CNTL, RD1, RD4, SENS, AUX2, AUX5, TDMCI
CHCALC	RD1, RD4, RD5, AUX2, AUX4, AUX5, CHLCC
TDSORT	none
OTAB1	CNTL, RD1, NCOS, AUX3, C1, C2, C3, C4, C5, C6, C7, C8, C9, C10
OTAB2	RD4, C2, C3, C4, C9
OTAB3	C2, C3, C4, C5, C6, C7, C8, C9, C10
OTAB4	RD3, NCOS, C10
OTAB5	RD4, AUX5
OSENS	CNTL, RD4, RD5, SENS, TDXUC, TDFR, TDK, TDUC, TDWOR, TDFPR, TDCMH, TDDMH, TDRM, TDSRTS, TDCRTS, TDDRTS, TDMCI

Figure 7: Correspondence Between Common Blocks and Subroutines
(Concluded)

4.15 Flow Chart

Figure 8 contains a detailed flow-chart of the computer program. On the left of the figure is the actual flow of the program from subroutine to subroutine. The comments on the right side of the figure describe the functions of the various subroutines.



SNSSET initializes sensitivity analysis variables.

One pass through the loop calculates a single LCC estimate.

If $ITER > 1$ PRMPT1 asks the user whether to reread the five "standard" data files. In any case, PRMPT1 asks the user the destination of the output and whether minimum or maximum prompting is to be used.

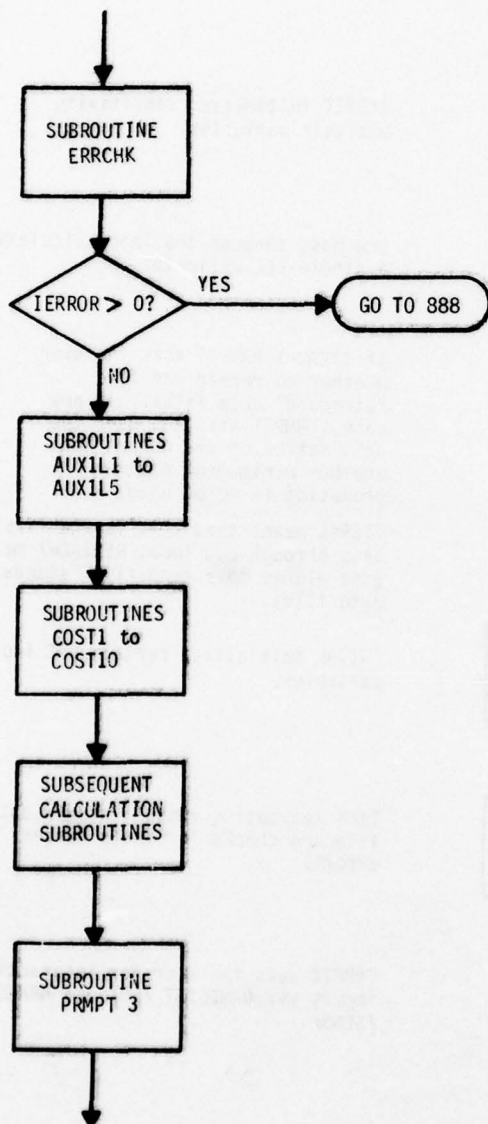
ITER=1 means that this is the first pass through LCC loop; REREAD=1 means user wishes to reread five "standard" data files.

INITAL initializes certain LCC input variables.

Each subroutine reads a single data file and checks for index number errors.

PRMPT2 asks the user for interactive inputs via NAMELIST /GO/ and NAMELIST /SENS/.

Figure 8: Flow-Chart of Computer Program (Page 1 of 3)



ERRCHK checks the inputs for fatal and non-fatal errors.

The LCC is not processed if ERRCHK finds any fatal errors.

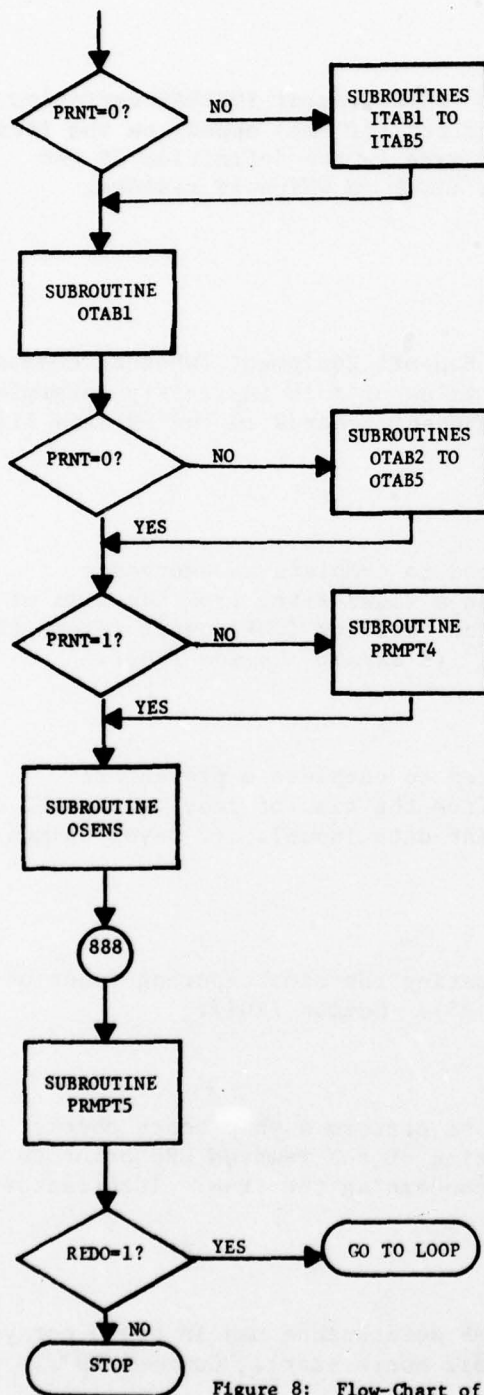
These subroutines calculate the auxiliary variables.

These subroutines calculate the ten cost elements.

Subroutine NONCOS calculates non-cost variables having to do with maintenance man-hours. The other subsequent calculation subroutines calculate sensitivity with respect to certain parameters.

PRMPT3 notifies the user that the LCC has been completed and allows the user to adjust the terminal to a new page.

Figure 8: Flow-Chart of Computer Program (Page 2 of 3)



If the user requested off-line output, print Input Tables 1 to 5 on the off-line printer.

OTAB1 prints the LCC summary output table at the terminal and/or on the off-line printer.

If the user requested off-line output, print Output Tables 2 to 5 on the off-line printer.

If the user requested output at the terminal, allow the user to adjust the terminal to a new page before the sensitivity analysis output.

OSENS prints the results of the sensitivity analysis at the terminal and/or on the off-line printer.

PRMP5 asks the user whether another run is desired.

REDO=1 means the user wants another run.

Figure 8: Flow-Chart of Computer Program (Page 3 of 3)

APPENDIX A: GLOSSARY

The following glossary defines all consequential FORTRAN variables. The name of the variable and its subscripts (if any) appear on the left. Below and to the right of the variable name is the definition of the variable and a reference to the common block in which it resides.

A

number of different types of Support Equipment (whether common or peculiar) at all levels [the value of A is implicitly determined by the LCC Model from the number of records in the SE data file]. Common /RD1/.

ADCM

average number of days required to complete an emergency corrective maintenance trip to a radar site, from the time of leaving the central maintenance facility (CMF) until return to the CMF. [Government data input] (3 days). Common /RD1/.

ADPM

average number of days required to complete a preventive maintenance trip to a site, from the time of leaving the CMF until return to the CMF. [Government data input]. (2 days) Common /RD1/.

B

the coefficient used in evaluating the stock sparing function F. [Government data input] (1.65). Common /RD1/.

BCMH(1)

average manhours at the site to perform a shop bench check, screening and fault verification of the removed LRU prior to initiating repair action or condemning the item. [Contractor data input]. Common /RD5/.

CAA

available working time per CMF maintenance man in hours per year [Government data input] (1512 hours/year). Common /RD1/.

CADRE

number of personnel from the AAC radar squadron who are designated to perform maintenance training over the lifetime of the system. [Government data input] (10). Common /RD1/.

CAS(1)

average number of LRUs of type i in the central maintenance facility (CMF) pipeline, i.e., the expected number of demands at the CMF on supply for LRU type i over the average CMF repair pipeline time. (Auxiliary variable). Common /AUX4/.

CBCMH(1)

average manhours at the CMF to perform a shop bench check, screening and fault verification of a removed LRU of type i. (This time is assumed to be the same for a falsely pulled LRU as for a failed LRU.) [Contractor data input] Common /RD5/.

CCF

the change in the value of CCOND(1) as computed in subroutines DWOR,DSRTS,DCRTS, and DDRTS. The parameter CCF is sent to subroutine CHCALC where the resulting change in LCC is computed. Common /CHLCC/.

CCLH(1)

average number of manhours expended by CMF personnel in corrective maintenance performed at the CMF on LRU type i per year. (Auxiliary variable) Common /AUX3/.

CCMP

average number of CMF personnel required for an emergency corrective maintenance trip to a radar site. [Contractor data input] Common /RD1/.

CCOND(1)

fraction of failed (and removed) LRUs which are condemned at the CMF, due to wear-out. [Auxiliary variable] Common /AUX1/.

CDR

daily labor rate for CMF personnel on a maintenance trip to a radar ste. [Government data input] (\$160 per day) Common /RD1/.

CDWH

working hours per day credited to each CMF personnel on a maintenance trip to a site. [Government data input] (8 hours) Common /RD1/.

CF

the change in the value of CRTS(1) as computed in subroutines DWOR,DSRTS,DCRTS, and DDRTS. The parameter CF is sent to subroutine CHCALC where the resulting change in LCC is computed. Common /CHLCC/.

CFCRTS(1)

change in CMF repair fraction (CRTS(1)) for LRU 1 from which the resulting change in LCC (TDCRTS(1)) is calculated. Common /TDCRTS/.

CHLCC

the Sensitivity Analysis variable that represents the change in total LCC that results from a fractional change in any of the factors SRTS(1), CRTS(1), DRTS(1), or WOR(1). This variable is computed internally in the Sensitivity Analysis component of the LCC Model in subroutine CHCALC and is passed back to the calling subroutine (DWOR,DSRTS,DCRTS, or DDRTS). Common /CHLCC/.

CLH

yearly total maintenance manhours expended by CMF personnel [Internally computed variable] Common /RD1/.

CLR

hourly labor rate at the CMF for radar maintenance personnel [Government data input] (\$13.97 per hour) Common /RD1/.

CMH(1)

average manhours to perform corrective maintenance at the CMF on a failed LRU of type 1, including fault isolation, repair and verification. [Contractor data input] Common /RD5/.

COC()

same as COS(), except refers to the CMF; not applicable if NSEC()=0. [Contractor data input] Common /RD2/.

COD()

same as COS() except refers to the depot; not applicable if NSED()=0. [Contractor data input] Common /RD2/.

COND(i)

fraction of failed (and removed) LRUs of type i which are condemned at the site. (Here we only allow either COND(i)=0 or COND(i)=1, depending on the repair level RL(i).) [Auxiliary variable] Common /AUX1/.

CONF

confidence factor (the probability that the stock of a given LRU type will last from one resupply trip to the next). [Government data input] (.98) Common /RD1/.

CONFLO

default value of CONF. If user submits a value for CONF that is less than CONFLO, a warning message is printed and CONF gets assigned the value of CONFLO. CONFLO is set to .84134 in subroutine INITIAL. Common /INIT/.

COS()

annual cost to operate and maintain one piece of Support Equipment of type at a site, expressed as a fraction of the unit cost (SEC()); not applicable if NSEC()=0. [Contractor data input] Common /RD2/.

CPMI

the shortest preventive maintenance interval for CMF personnel trips to a radar site (in hours). [Contractor data input] Common /RD1/.

CPMP

average number of CMF personnel required on a scheduled preventive maintenance trip to a radar site. [Contractor data input] Common /RD1/.

CPWT

manpower cost per unit weight (pound) affected by enhancement (or fixup), based upon an aggregate to include both hardware ECPs and the physical aspects of the reprogramming of firmware. [Contractor data input] Common /RD1/.

CRCT

CMF repair cycle time (in weeks) for an LRU from removal at the site of the failed (or falsely pulled) item until it is returned to CMF serviceable stock. [Government data input] (1.43 weeks) Common /RD1/.

CRTS(1)

fraction of failed (and removed) LRUs of type 1 which are repaired at the CMF. [Auxiliary variable] Common /AUX1/.

CRTS1(1)

the fraction of failed LRUs of type 1 which are to be repaired at the CMF under repair level RL(1)=1. [Contractor data input] Common /RD5/.

CRTS2(1)

the fraction of failed LRUs of type 1 which are to be repaired at the CMF under repair level RL(1)=2. [Contractor data input] Common /RD5/.

C1

total cost of Cost Element 1. Common /C1/.

C2

total cost of Cost Element 2. Common /C2/.

C2C

part of Cost Element 2 attributable to the CMF. Common /C2/.

C2D

part of Cost Element 2 attributable to the depot. Common /C2/.

C2L(1)

part of Cost Element 2 attributable to LRU type 1. Common /C2/.

C2S

part of Cost Element 2 attributable to the radar sites. Common /C2/.

C3

total cost of Cost Element 3. Common /C3/.

C3C

part of Cost Element 3 attributable to the CMF. Common /C3/.

C3D

part of Cost Element 3 attributable to the depot. Common /C3/.

C3L(1)

part of Cost Element 3 attributable to LRU type 1. Common /C3/.

C3S

part of Cost Element 3 attributable to the radar sites. Common /C3/.

C4

total cost of Cost Element 4. Common /C4/.

C4I

initial part of Cost Element 4. Common /C4/.

C4L(1)

part of Cost Element 4 attributable to LRU type 1. Common /C4/.

C4R

recurring part of Cost Element 4. Common /C4/.

C5

total cost of Cost Element 5. Common /C5/.

C6

total cost of Cost Element 6. Common /C6/.

C6C

part of Cost Element 6 attributable to the CMF. Common /C6/.

C6D

part of Cost Element 6 attributable to the depot. Common /C6/.

C6I

initial part of Cost Element 6. Common /C6/.

C6R

recurring part of Cost Element 6. Common /C6/.

C6S

part of Cost Element 6 attributable to the radar sites. Common /C6/.

C7

total cost of Cost Element 7. Common /C7/.

C8

total cost of Cost Element 8. Common /C8/.

C8C

part of Cost Element 8 attributable to the CMF. Common /C8/.

C8D

part of Cost Element 8 attributable to the depot. Common /C8/.

C8I

initial part of Cost Element 8. Common /C8/.

C8R

recurring part of Cost Element 8. Common /C8/.

C8S

part of Cost Element 8 attributable to the radar sites. Common /C8/.

C9

total cost of Cost Element 9. Common /C9/.

C9C

part of Cost Element 9 attributable to the CMF. Common /C9/.

C9D

part of Cost Element 9 attributable to the depot. Common /C9/.

C9I

initial part of Cost Element 9. Common /C9/.

C9L(i)

part of Cost Element 9 attributable to LRU type i. Common /C9/.

C9R

recurring part of Cost Element 9. Common /C9/.

C9S

part of Cost Element 9 attributable to the the radar sites.
Common /C9/.

C10

total cost of Cost Element 10. Common /C10/.

C10HW(j)

hardware part of Cost Element 10 which is attributable to
functional area j. Common /C10/.

C10I

initial part of Cost Element 10. Common /C10/.

C10R

recurring part of Cost Element 10. Common /C10/.

C10SW(j)

software part of Cost Element 10 which is attributable to
functional area j. Common /C10/.

DAA

available working time per depot maintenance man in hours per year. [Government data input] (1512 hours/year) Common /RD1/.

DAS(i)

average number of LRUs of type i in the depot pipeline, more precisely, the average number of repairable failures of LRU type i which are received by the depot over its average repair cycle time (DRCT). [Auxiliary variable] Common /AUX4/.

DBCMH(i)

average manhours at the depot to perform a shop bench check, screening and fault verification of a failed LRU of type i. [Contractor data input] Common /RD5/.

DCF

the change in the value of DCOND(i) as computer in subroutines DWOR,DSRTS,DCRTS, and DDRTS. The parameter DCF is sent to subroutine CHCALC where the resulting change in LCC is computed. Common /CHLCC/.

DCLH(i)

average number of hours expended by depot personnel in corrective maintenance on LRU type i per year. [Auxiliary variable] Common /AUX3/.

DCOND(i)

fraction of failed (and removed) LRUs of type i which are condemned at the depot, due to wear-out. [Auxiliary variable] Common /AUX1/.

DF

the change in the value of DRTS(i) as computed in subroutines DWOR,DSRTS,DCRTS, and DDRTS. The parameter is sent to subroutine CHCALC where the resulting change in LCC is computed. Common /CHLCC/.

DFDRTS(i)

change in depot repair fraction for LRU type i from which the resulting change in LCC (TDDRTS(i)) is calculated. Common /TDDRTS/.

DLH

average yearly number of manhours expended by depot personnel in corrective maintenance (total for all LRUs). [Auxiliary variable] Common /AUX3/.

DLR

hourly labor rate at the depot. [Government data input] (\$15.52 per hour) Common /RD1/.

DMH(1)

average manhours to perform corrective maintenance at the depot on a failed LRU of type i, including fault isolation, repair and verification. [Contractor data input] Common /RD5/.

DRCT

depot repair cycle time, i.e., the time from when the CMF sends a repairable LRU to the depot until it is repaired and placed in depot serviceable stock (in weeks). [Government data input] (6 weeks) Common /RD1/.

DRTS(1)

fraction of failed (and removed) LRUs of type i which are repaired at the depot. [Auxiliary variable] Common /AUX1/.

DRTS1(1)

the fraction of failed LRUs of type i which are to be repaired at the depot (Sacramento ALC) under repair level RL(1)=1. [Contractor data input] Common /RD5/.

DRTS2(1)

the fraction of failed LRUs of type i which are to be repaired at the depot under repair level RL(1)=2. [Contractor data input] Common /RD5/.

DRTS3(1)

the fraction of failed LRUs of type i which are to be repaired at the depot under repair level RL(1)=3. [Contractor data input] Common /RD5/.

DS

fraction of radar system incidents (demands for maintenance action) detected automatically, including those met by repair in place, and those which do not cause system to become inoperative (e.g., failure of a redundant LRU). [Contractor data input] Common /RD1/.

DSTK(i)

the average (fractional) increase in the number of site spares (STK(i)) of LRU type i which would result from a unit increase in the computed value of SAS(i). DSTK(i) is used in Sensitivity Analysis calculations. [Auxiliary variable] Common /AUX5/.

ECMT

the average number of emergency corrective maintenance trips from the CMF to the radar sites per year (total for all sites). [Auxiliary variable] Common /AUX5/.

ENYR(j)

expected number of enhancements (including fixups) per year which will occur in Functional Area j throughout the life of the system. [Government data input] (1 per Functional Area per year) Common /RD3/.

EXIT

internal flag which is set to 1 when user communicate to program via the prompting routines that he or she wishes to skip to the end of the program. Common /CNTL/.

FA

number of different functional areas in each the radar. [Contractor data input] Common /RD3/.

FAC()

cost of new facility number which is required at every radar site (see Section C.8). [Contractor data input] Common /RD1/.

FANAME(.,.)

nomenclature for a given functional area type. Common /RD3/.

FCS

cost for power at radar sites, in dollars per kilowatt hour per site. [Government data input] (\$0.0518/KW hr/site) Common /RD1/.

FINC

the fractional increase in those LCC factors for which Sensitivity Analysis is provided. Appears in all Sensitivity Analysis computations. [Contractor interactive input] Common /SENS/.

FPR(1)

false pull rate for LRU type 1, i.e., that multiple of failed LRUs which are removed but haven't failed (in most part this factor is to allow for the inability of fault isolation to a single LRU). (Thus (# of removed LRUs)=(1+FPR(1))(# of failed LRU).) [Contractor data input] Common /RD5/.

FR(1)

the failure rate (maintenance incident rate) of LRU type 1. Sensitivity Analysis is provided for this factor. This factor is computed internally by the LCC Model in units of failures per million hours, i.e., $FR(1) = (10E6)/MTBI(1)$. Not a program variable.

FSD

total contract cost to the Government for Phase II (Full-Scale Development). This is a lump-sum term in the Cost Element for Development & Production. [Contractor data input] Common /RD1/.

H

average transportation cost per site of a maintenance trip from the CMF to a site. [Government data input] (\$1600 per trip) Common /RD1/.

HWCF(j)

hardware change fraction for Functional Area j, the expected fraction of the hardware or firmware component of Functional Area j which is to be changed per enhancement (or fixup) required in that functional area. [Contractor data input] Common /RD3/.

IDCMH(i)

holds index numbers of LRUs sorted with respect to the values in array TDCMH. Common /TDCMH/.

IDCRTS(i)

holds index numbers of LRUs sorted with respect to the values in array TDCRTS. Common /TDCRTS/.

IDDMH(i)

holds index numbers of LRUs sorted with respect to the values in array TDDMH. Common /TDDMH/.

IDDRTS(i)

holds index numbers of LRUs sorted with respect to the values in array TDDRTS. Common /TDDRTS/.

IDFPR(i)

holds index numbers of LRUs sorted with respect to the values in array TDFPR. Common /TDFPR/.

IDFR(i)

holds index numbers of LRUs sorted with respect to the values in array TDFR. Common /TDFR/.

IDMCI(i)

holds index numbers of LRUs sorted with respect to the values in array TDMCI. Common /TDMCI/.

IDRM(i)

holds index numbers of LRUs sorted with respect to the values in array TDRM. Common /TDRM/.

IDSRTS(i)

holds index numbers of LRUs sorted with respect to the values in array TDSRTS. Common /TDSRTS/.

IDWOR(i)

holds index numbers of LRUs sorted with respect to the values in array TDWOR. Common /TDWOR/.

IDUC(1)

holds index numbers of LRUs sorted with respect to the values in array TDUC. Common /TDUC/.

IERRI

number of index number errors found in the data files. Not in a common block. Local to MAIN routine.

IERROR

counter for the number of fatal errors which are found among the values in the user's data. Common /ERROR/.

IMC

initial management cost to introduce a new line item of supply into the Air Force inventory. [Government data input] (\$40.91) Common /RD1/.

IMH(1)

average manhours to perform in place repair of LRU type 1, including fault isolation, repair and verification. [Contractor data input] Common /RD5/.

INO(1)

array holding the index numbers found in the LRU equipment input file (unit 14). Common /RD4/.

INO15(1)

array holding the index numbers found in the LRU maintenance input file (unit 15). Common /RD5/.

ITER

counts the number of passes (i.e., reruns) through the LCC loop. Common /CNTL/.

IWARN

counter for the number of non-fatal errors which are found in the user's data. Common /ERROR/.

K

the reliability factor which converts predicted failure rates (incident rates) to operational failure rates (incident rates). [Government data input] (3) Common /RD1/.

LTEMP

set to 1 if off-line printing is desired and LDERV does not equal zero. Not in a common block. Local to MAIN routine.

M

number of radar sites in the SEEK IGLOO System. [Government data input] (13) Common /RD1/.

MAXFA

maximum number of functional area types allowed. Common /INIT/.

MAXLR

maximum number of LRU types allowed. Common /INIT/.

MAXSE

maximum number of support equipment types allowed. Common /INIT/.

MAXPMT

set to 1 when user chooses to have a complete set of program-generated prompts. Common /CNTL/.

MCF(1)

"mission critical flag", identifies those LRUs which given rise to an emergency corrective maintenance trip by CMF personnel on failure. Specifically, MCF(1)=1 if LRU type 1 is "mission critical", but is neither stocked on site (i.e., STK(1)=0) nor redundant (i.e., QR(1)=0); and MCF(1)=0 otherwise. [Auxiliary variable] Common /AUX5/.

MCI(i)

"mission critical indicator": serves to identify those LRU types which are "mission critical" (i.e., those LRUs whose failure will cause system failure). In addition, for mission critical LRUs, the value of MCI(i) will indicate one of two possible site sparing strategies. [Contractor data input]. More precisely, MCI(i) should be set equal to:

0 if LRU type i is not mission critical

1 if LRU type i is mission critical, but no special site sparing provision made for it

2 if LRU type i is mission critical and provision is made for at least one spare of LRU type i to be stocked at each radar site.

Common /RD4/.

MCIC(i)

change in mission critical indicator for LRU i from which the expected change in LCC (TDMCI(i)) is calculated. Note that MCIC(i) can assume values 0,1, and -1 only. Common /TDMCI/.

MCRS

miscellaneous consumption rate per radar site, to include all consumables except fuel in \$/year (under the assumption that this consumption rate is invariant under differing system designs); lamps, fuses, and other items which are of lower indenture level than the LRU and not included in any RM(i) factor must be included here. [Contractor data input] Common /RD1/.

MMH(j)

average manhours required to perform scheduled preventive maintenance by site personnel in Functional Area j. [Contractor data input] Common /RD3/.

MTBI(i)

mean time between incidents (maintenance actions) for LRU type i, in hours. [Contractor data input] Common /RD4/.

N

largest index number which is found in LRU equipment inputs file (unit 14). Common /RD4/.

N15

largest LRU index number found on LRU maintenance inputs file
(unit 15 file). Common /RD5/.

NCP

number of CMF radar maintenance personnel.
[Contractor data input] Common /RD1/.

NDP

number of depot personnel required for maintenance of the SEEK
IGLOO System. [Contractor data input] Common /RD1/.

NSEC()

number of pieces of Support Equipment of type required at the
CMF (whether common or peculiar). [Contractor data input] Common
/RD2/.

NSED()

number of pieces of peculiar Support Equipment of type required
at the depot. [Contractor data input] Common /RD2/.

NSES()

number of pieces of Support Equipment of type required at each
site (whether common or peculiar). [Contractor data input] Common
/RD2/.

NSP

number of radar maintenance personnel per radar site. [Contractor
data input] Common /RD1/.

OST

average order and shipping time (in weeks) from the depot to the
CMF. [Government data input] (1.56 weeks) Common /RD1/.

PA(i)

number of new "P" coded (i.e., procurable) repairable assemblies,
subassemblies, and piece parts which will be stocked to support
repair of LRU type i. [Contractor data input] Common /RD4/.

PAMH(1)

average manhours expended on a radar for Preparation and Access to a failed LRU of type 1. [Contractor data input] Common /RD5/.

PIUP

number of operating years of the SEEK IGLOO Radar System. [Government data input] (20 years) Common /RD1/.

PME

Phase III CWBS, level 2 item "Prime Mission Equipment", covering all (12) production systems (including learning-curve effects), and excluding level 3 items "Software" and "Refurbishment". This is a lump-sum term in the Cost Element for Development & Production. [Contractor data input] Common /RD1/.

PMFA(j)

annual preventive maintenance man-hours expended at each radar site within Functional Area j. Common /NCOS/.

PPRS

consumption rate for primary power (for prime mission equipment) per radar site, in kilowatts. [Contractor data input] Common /RD1/.

PRNT

print flag: 0 means user wants terminal (unit 6) printing only; 1 means user wants off-line (unit 7) printing only; 2 means user wants both. If user requests neither terminal printing nor offline printing, the user gets terminal output anyway. Common /CNTL/.

PRODX

sum of all Phase III CWBS level 2 items excepting "Prime Mission Equipment," "Training," "Support Equipment," and "Initial Spares and Repair Parts." This is a lump-sum term in the Cost Element for Development & Production. [Contractor data input] Common /RD1/.

QPA(1)

number of LRUs of type 1 in each radar. [Contractor data input] Common /RD4/.

QR(i)

number of LRUs of type i which are redundant in each radar, i.e., a minimum of $(QPA(i) - QR(i))$ are sufficient to maintain operational status of the system. [Contractor data input] Common /RD4/.

REDO

set to 1 after LCC calculation if user chooses to have yet another LCC calculation. Common /CNTL/.

REFURB

Phase III CWBS level 3 item "Refurbishment" covering refurbishment of 2 preproduction options. This is a lump-sum term in the Cost Element for Development & Production. [Contractor data input] Common /RD1/.

REREAD

(only applicable if on second or subsequent pass of the LCC) set to 1 when user chooses to reread the five "standard" data files. Common /CNTL/.

RIP(i)

fraction of failures (maintenance incidents) of LRU type i which are met by repair in place on the radar. [Contractor data input] Common /RD5/.

RL(i)

repair level for LRU type i, equals 0 if LRU is to be condemned on failure (at the site), equals 1 if LRU repair is to be attempted (i.e., lowest level repair) at the site, equals 2 if LRU repair is to be first attempted at the CMF and equals 3 if LRU type i is to be repaired only at the depot level. [Contractor data input] Common /RD5/.

RM(i)

repair materials factor for LRU type i, the fraction of UC(i), unit cost that is consumed (in lower indenture level components of the LRU) in the repair of LRU type i. [Contractor data input] Common /RD4/.

RMC

recurring annual cost to maintain an item of supply in the depot inventory system. [Government data input] (\$104.20 per year) Common /RD1/.

RMH(i)

average manhours to fault isolate, remove and replace a faulty (or falsely pulled) LRU of type i and verify restoration of the system to operational status. [Contractor data input] Common /RD5/.

SA

annual cost to maintain a line item in the site or CMF inventory system. [Government data input] (\$20.20) Common /RD1/.

SAA

available work time per site radar maintenance man in hours per year. [Government data input] (1512 hours per year) Common /RD1/.

SAS(i)

average number of failed and falsely pulled LRUs of type i which have accumulated at a site by the end of an order and shipping interval (SOSI). [Auxiliary variable] Common /AUX4/.

SCF

the change in the value of COND(i) as computed in subroutines DWOR,DSRTS,DCRTS, and DDRTS. The parameter is sent to subroutine CHCALC where the resulting change in LCC is computed. Common /CHLCC/.

SEC()

unit cost of a piece of Support Equipment of type . [Contractor data input] Common /RD2/.

SENAME(.,.)

nomenclature for a given support equipment type. Common /RD2/.

SF

the change in the value of SRTS(i) as computed in subroutines DWOR, DSRTS, DCRTS, and DDRTS. The parameter is sent to subroutine CHCALC where the resulting change in LCC is computed. Common /CHLCC/.

SFSRTS(i)

change in site repair fraction for LRU i from which the expected change in LCC (TDSRTS(i)) is calculated.

SIZE(j)

size of the software component of Functional Area j, including all programs residing in firmware, in number of object instructions (or the equivalent, for higher order languages). [Contractor data input] Common /RD3/.

SMH(i)

average manhours to perform site-level corrective maintenance of a removed LRU of type i, including fault isolation, repair and verification. [Contractor data input] Common /RD5/.

SMI(j)

average scheduled maintenance interval (in hours) for preventive maintenance of Functional Area j by site maintenance personnel. [Contractor data input] Common /RD3/.

SMMH

yearly manhours expended by site radar maintenance personnel in performing on-radar maintenance. [Internally computed variable in Section C.3.2] Common /NCOS/.

SMTBI

overall mean time between incidents (maintenance actions) for one radar, in hours; must satisfy

to reflect the fact that a radar is more than the serial array of its component LRUs. [Contractor data input] Common /RD1/.

SNRTS(i)

fraction of failed (and removed) LRUs of type i which are sent to the CMF (or depot) for possible repair. [Auxiliary variable] Common /RD1/.

SOSI

order and shipping interval between each site and the CMF in weeks (i.e., number of weeks between resupply trips). [Government data input] (1 week) Common /RD1/.

SPMH

yearly manhours expended by site radar maintenance personnel in performing corrective maintenance on LRUs off the radar (i.e., not including on-radar manhours). [Internally computed variable in Section C.3.2] Common /NCOS/.

SPRS

consumption rate for secondary power (heating and air conditioning necessary for PME) per radar site, in kilowatts. [Contractor data input] Common /RD1/.

SRCT

average site repair cycle time in weeks. (The elapsed time from removal of a failed LRU until it is repaired at the site and placed in site serviceable stock.) [Government data input] (.43 weeks)

SRTS(1)

fraction of failed (and removed) LRUs of type 1 which are repaired at the site. [Auxiliary variable] Common /AUX1/.

SRTS1(1)

the fraction of failed LRUs of type 1 which are to be repaired at the radar site under repair level RL(1)=1. [Contractor data input] Common /RD5/.

STK(1)

the actual number of off-equipment spares of LRU type 1 per site as determined from STK1(1) with a possible adjustment for the mission critical indicator MCI(1) (i.e., $STK(1) = \text{Max} [STK1(1), MCI(1) - 1]$). [Auxiliary variable] Common /AUX5/.

STK1(i)

the tentative number of off-equipment spares of LRU type i per site, including safety stock, as determined by the site stocking criterion and the confidence level CONF, but before any adjustment is made for the mission critical indicator MCI(i). [Auxiliary variable] Common /AUX5/.

STMH

sum of SMMH and SPMH. Common /NCOS/.

SW

Phase III CWBS level 3 item "Software" covering 12 production systems. This is a lump-sum term in the Cost Element for Development & Production. [Contractor data input] Common /RD1/.

SWCF(j)

software change fraction for Functional Area j, the expected fraction of the software component of Functional Area j (including programs residing in firmware) which is to be changed per enhancement (or fixup) required in that functional area. [Contractor data input] Common /RD3/.

SWFAC

one-time cost of a software maintenance facility, if one is intended to be used, exclusive of the fixed and variable costs SWFIX(j); and SWVAR(j). [Contractor data input] Common /RD1/.

SWFIX(j)

the fixed component of manpower cost per software enhancement (SWCOST(j)) in Functional Area j when described as a function of the number of object instructions programmed (or reprogrammed) in thousands of dollars. [Contractor data input] Common /RD3/.

SWVAR(j)

the coefficient of the variable component of manpower cost per software enhancement (SWCOST(j)) in Functional Area j when described as a function of the number of object instructions programmed (or reprogrammed) in dollars. [Contractor data input] Common /RD3/.

TC

average cost per student per week to offer site, CMF, and depot maintenance training (not including student salaries). [Contractor data input] Common /RD1/.

TCCAD

cost per student per week to offer "CADRE" training (not including student salaries). [Contractor data input] Common /RD1/.

TDCMH(1)

value of sensitivity of LCC with respect to CMF maintenance hours for LRU type 1. Common /TDCMH/.

TDCRTS(1)

value of sensitivity of LCC with respect to CMF repair fraction for LRU type 1. Common /TDCRTS/.

TDDMH(1)

value of sensitivity of LCC with respect to depot maintenance hours for LRU type 1. Common /TDDMH/.

TDDRTS(1)

value of sensitivity of LCC with respect to depot repair fraction for LRU type 1. Common /TDDRTS/.

TDFPR(1)

value of sensitivity of LCC with respect to the false pull rate for LRU type 1. Common /TDFPR/.

TDFR(1)

value of sensitivity of LCC with respect to failure rate for LRU type 1. Common /TDFR/.

TDK

value of global sensitivity of LCC with respect to the K factor. Common /TDK/.

TDMCI(1)

value of sensitivity of LCC with respect to mission critical indicator for LRU type 1. Common /TDMCI/.

TDRM(i)

value of sensitivity of LCC with respect to the repair materials factor for LRU type i. Common /TDRM/.

TDSRTS(i)

value of sensitivity of LCC with respect to site repair fraction for LRU type i. Common /TDSRTS/.

TDUC(i)

value of sensitivity of LCC with respect to unit cost for LRU type i. Common /TDUC/.

TDWOR(i)

value of sensitivity of LCC with respect to wear out rate for LRU type i. Common /TDWOR/.

TDXUC

value of global sensitivity of LCC with respect to XUC factor. Common /TDXUC/.

TE

cost of peculiar training equipment required by the Government; not to include any contractor-incurred costs already accounted for in development costs within the Cost Element for Development & Production. [Contractor data input] Common /RD1/.

TITL(.)

array that holds the (up to) 27 character title for a particular LCC calculation. Common /TITL/.

TR

average turnover rate for site, CMF, and depot maintenance personnel. [Government data input] (1.0) Common /RD1/.

TW

average number of weeks training required for site, CMF, and depot maintenance personnel. [Contractor data input] Common /RD1/.

TWCAD

number of weeks required for the initial training of "CADRE" personnel. [Contractor data input] Common /RD1/.

UC(i)

unit cost to the Government of LRU type i, final production item on the cumulative average learning curve, as used to calculate the value of PME in the Cost Element for Development & Production. (Note: Unit production cost, G&A, and profit are to be included.) [Contractor data input] Common /RD4/.

VAL

total contract cost to the Government for Phase I (Design Validation). This is a lump-sum term in the Cost Element for Development & Production. [Contractor data input] Common /RD1/.

WEIGHT(j)

weight of all hardware and firmware in Functional Area j, in pounds. [Contractor data input] Common /RD3/.

WFR(i)

average number of failed (and removed) LRUs of type i per week (total for all sites). [Auxiliary variable] Common /AUX2/.

WOR(i)

the wear-out rate of a normally repairable LRU of type i. The value of WOR(i) is computed internally by the LCC Model where WOR(i) equals either WOR1(i), WOR2(i), or WOR3(i), depending on the value of the repair level RL(i). Sensitivity Analysis is performed by the LCC Model on WOR(i). Not an actual program variable.

WOR1(i)

the wear-out rate of a (normally repairable) LRU of type i under repair level RL(i)=1. [Contractor data input] Common /RD5/.

WOR2(i)

the wear-out rate of a (normally repairable) LRU of type i under repair level RL(i)=2. [Contractor data input] Common /RD5/.

WOR3(i)

the wear-out rate of a (normally repairable) LRU of type i under repair level RL(i)=3. [Contractor data input] Common /RD5/.

WR(i)

the fraction of failed LRUs of type i which are discarded, either through a "discard-on-failure" decision (i.e., COND(i)=1) or through wear-out of normally-repairable LRUs. [Auxiliary variable] Common /AUX1/.

XUC

dummy cost adjustment factor. XUC is provided in the LCC Model to facilitate LCC tradeoffs involving any factors (such as the use of high-reliability parts throughout the design) which can be shown to multiply baseline costs of all LRU's and PME by a single constant. The value of XUC must be set at unity (1.0) in any runs of the model whose outputs correspond to proposed or contractual baseline costs. [Government data input] (1.0) Common /RD1/.

YFR(i)

average number of failed (and removed) LRUs of type i per year (total for all radar sites). [Auxiliary variable] Common /AUX2/.

YOH

yearly operating hours of the radar system. [Government data input] (8766 hours per year) Common /RD1/.

YSCAD

yearly salary for "CADRE" personnel. [Government data input] (\$13400) Common /RD1/.

YSLR

yearly labor rate for site radar maintenance men. [Government data input] (\$52000 per man per year) Common /RD1/.

Z

the number of standard deviations which corresponds to the CONF value, i.e., the value Z is determined so that the area under the standard normal curve to the left of the value Z equals CONF. [Auxiliary variable] Common /AUX5/.

APPENDIX B: PROGRAM LISTING

```

C*****00000010
C*      SEEK IGLOO LIFE CYCLE COST MODEL      *00000020
C* THIS PROGRAM IS DESIGNED TO RUN REPETITIVE LIFE CYCLE COST *00000030
C* CALCULATIONS IN AN INTERACTIVE ENVIRONMENT. THERE ARE SIX INPUT *00000040
C* DATA SETS: *00000050
C*   UNIT 5 - RECEIVES INPUTS TYPED BY THE USER AT THE TERMINAL. *00000060
C*   UNIT 11 - MISCELLANEOUS INPUTS (MAINLY SCALARS). *00000070
C*   UNIT 12 - INPUTS BY SUPPORT EQUIPMENT TYPE. *00000080
C*   UNIT 13 - INPUTS BY FUNCTIONAL AREA. *00000090
C*   UNIT 14 - INPUTS BY LRU TYPE (LRU FILE 1). *00000100
C*   UNIT 15 - INPUTS BY LRU TYPE (LRU FILE 2). *00000110
C* TWO FILES RECEIVE THE OUTPUT: *00000120
C*   UNIT 6 - OUTPUT THAT GOES TO THE TERMINAL. *00000130
C*   UNIT 7 - OUTPUT THAT GOES TO THE OFF-LINE PRINTER. *00000140
C*****00000150
C      00000160
C*****00000170
C*      COMMON BLOCK ORGANIZATION *00000180
C* IN GENERAL, FOR EVERY SUBROUTINE THERE IS A COMMON BLOCK THAT *00000190
C* CONTAINS ALL OF THE VARIABLES THAT FIRST RECEIVED VALUES IN THAT *00000200
C* SUBROUTINE. HENCE, ALL VARIABLES THAT ARE READ FROM THE UNIT 11 *00000210
C* FILE IN SUBROUTINE READ1 WILL BE IN COMMON /RD1/; ALL VARIABLES *00000220
C* CALCULATED IN SUBROUTINE AUXIL1 WILL BE IN COMMON /AUX1/; ALL *00000230
C* VARIABLES CALCULATED IN SUBROUTINE COST1 WILL BE IN COMMON /C1/; *00000240
C* ALL VARIABLES CALCULATED IN SUBROUTINE DFR WILL BE IN COMMON/DFR/. *00000250
C*****00000260
C      00000270
C      COMMON /CNTL/ EXIT,ITER,MAXPNT,PRNT,REDO,REREAD
C      INTEGER EXIT,PRNT,REDO,REREAD      00000280
C      00000290
C      00000300
C      COMMON /TITL/ TITL(28)      00000310
C      00000320
C      COMMON /ERROR/ IERROR,IWARN      00000330
C      00000340
C      COMMON /INIT/ CONFLO,MAXPA,MAXLR,MAXSF      00000350
C      00000360
C      COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,CONF,CPI,
C      + CPMP,CPWT,CRCT,DAA,DLR,DPCT,DS,      00000370
C      + PAC(10),FCS,FSD,H,INC,K,M,MCRS,MXHS,      00000380
C      + NCP,NDP,NSP,OST,PIUP,PNE,PPRS,PRODX,REFURB,RMC,      00000390
C      + SA,SAA,SMTEI,SOST,SPRS,SRCT,SW,SWFAC,      00000400
C      + TC,TCCAD,TF,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR      00000410
C      INTEGER CADRE      00000420
C      REAL INC,K,MCRS,MXHS      00000430
C      00000440
C      00000450
C      COMMON /RD2/ A,COC(150),COD(150),COS(150),INOSE(150),NSEC(150),
C      + NSED(150),NSPS(150),SEC(150),SENAME(150,24)      00000460
C      REAL NSEC,NSED,NSES      00000470
C      INTEGER A      00000480
C      00000490
C      00000500
C      COMMON /RD3/ ENYR(10),PA,PANAME(10,24),HWCF(10),INOFA(10),MMH(10),
C      + SIZE(10),SMI(10),SWCF(10),SWFIX(10),SWVAR(10),
C      + WEIGHT(10)      00000510
C      00000520
C      00000530
C      INTEGER PA      00000540
C      REAL MMH      00000550
C      00000560
C      00000570
C      COMMON /RD4/ INO(200),LPNAME(200,24),LRPART(200,12),MCI(200),
C      + MTBI(200),N,PA(200),OPA(200),QR(200),RH(200),
C      + UC(200)      00000580
C      00000590
C      INTEGER OPA,QR      00000600
C      REAL MTBI      00000610
C      00000620
C      00000630
C      COMMON /RD5/ BCMH(200),CBCMH(200),CMH(200),CRTS1(200),CRTS2(200),

```

	+	DBCMMH(200), DMH(200), DRTS1(200), DRTS2(200),	00000640
	+	DRTS3(200), FPR(200), IMH(200), INO15(200), N15,	00000650
	+	PAMH(200), PIP(200), RL(200), RMH(200), SMH(200), SRTS1(200),	00000660
	+	WOR1(200), WOR2(200), WOR3(200)	00000670
		REAL IMH	00000680
		INTEGER RL	00000690
C		COMMON /SENS/ FINC, LDCMH, LDCRTS, LDDMH, LDDRTS, LDERV, LDFPR, LDFR,	00000700
	+	LDMCI, LDRH, LDSRTS, LDWOR, LDUC	00000710
C		COMMON /AUX1/ CCOND(200), COND(200), CRTS(200), DCOND(200),	00000720
	+	DRTS(200), SNRTS(200), SRTS(200), WR(200)	00000730
C		COMMON /AUX2/ YPR(200), WPR(200)	00000740
C		COMMON /AUX3/ CCLH(200), DCLH(200), DLH	00000750
			00000760
C		COMMON /AUX4/ CAS(200), DAS(200), SAS(200)	00000770
			00000780
C		COMMON /AUX5/ DSTK(200), ECMT, MCF(200), STK(200), STK1(200), Z	00000790
		INTEGER STK, STK1	00000800
C		COMMON /C1/ C1	00000810
			00000820
C		COMMON /C2/ C2, C2C, C2D, C2L(200), C2S	00000830
			00000840
C		COMMON /C3/ C3, C3C, C3D, C3L(200), C3S	00000850
			00000860
C		COMMON /C4/ C4, C4I, C4L(200), C4R	00000870
			00000880
C		COMMON /C5/ C5	00000890
			00000900
C		COMMON /C6/ C6, C6C, C6D, C6I, C6R, C6S	00000910
			00000920
C		COMMON /C7/ C7	00000930
			00000940
C		COMMON /C8/ C8, C8C, C8D, C8I, C8R, C8S	00000950
			00000960
C		COMMON /C9/ C9, C9C, C9D, C9I, C9L(200), C9R, C9S	00000970
			00000980
C		COMMON /C10/ C10, C10HW(10), C10I, C10R, C10SW(10)	00000990
			00001000
C		COMMON /NCOS/ CLH, PHFA(10), SHMH, SPNH, STMH	00001010
			00001020
C		COMMON /TDXUC/ TDXUC	00001030
			00001040
C		COMMON /TDFR/ IDFR(200), TDFR(200)	00001050
			00001060
C		COMMON /TDK/ TDK	00001070
			00001080
C		COMMON /TDUC/ IDUC(200), TDUC(200)	00001090
			00001100
C		COMMON /TDW/ TDW	00001110
			00001120
C		COMMON /TDWOR/ IDWOR(200), TDWOR(200), WP(200)	00001130
			00001140
C		COMMON /TDFPR/ IDFPR(200), TDFPR(200)	00001150
			00001160
C		COMMON /TDCMH/ IDCMH(200), TDCMH(200)	00001170
			00001180
C		COMMON /TDDMH/ IDDMH(200), TDDMH(200)	00001190
			00001200
C		COMMON /TDRH/ IDRH(200), TDRH(200)	00001210
			00001220
C		COMMON /TDSRTS/ IDSRTS(200), SPSRTS(200), TDSRTS(200)	00001230
			00001240
			00001250
			00001260


```

C                                00001270
C      COMMON /TDCRTS/ CFCRTS (200), IDCRTS (200), TDCRTS (200)      00001280
C                                00001290
C      COMMON /TDDRIS/ DFCRTS (200), IDDRIS (200), TDDRIS (200)      00001300
C                                00001310
C      COMMON /TDMCI/ IDMCI (200), MCIC (200), TDMCI (200)          00001320
C                                00001330
C      COMMON /CHLCC/ CCF, CF, CHLCC, DCF, DF, SCP, SF              00001340
C                                00001350
C                                00001360
C      1  FORMAT (1X/49H PROGRAM WILL TERMINATE AFTER NAMELIST INPUTS DUE, 00001370
C          *      3CH TO ERRORS IN THE INPUT FILES.)                00001380
C      2  FORMAT (1X/10H TOTAL OF ,I3,24H INPUT WARNING MESSAGES.)    00001390
C      3  FORMAT (1X/1X,26H PROGRAM TERMINATES DUE TO ,I3,14H INPUT ERRORS.) 00001400
C                                00001410
C                                00001420
C*****00001430
C*                                SNSSET                             *00001440
C*  INITIALIZE SENSITIVITY ANALYSIS VARIABLES AT THE START OF THE RUN. *00001450
C*****00001460
C                                00001470
C      CALL SNSSET                                                    00001480
C                                00001490
C*****00001500
C*                                LOOP ON ITER FOR EACH RUN OF THE LCC. *00001510
C*****00001520
C                                00001530
C      DO 999 ITER=1,50                                              00001540
C                                00001550
C*****00001560
C*                                PRMPT1                             *00001570
C*  CALL THE FIRST OF THE FIVE PROMPTING ROUTINES TO DETERMINE:      *00001580
C*  1- MAXPMT.                                                       *00001590
C*    MAXPMT=0 - USER WANTS MINIMUM PROMPTING (WITHOUT EXPLANATORY *00001600
C*              MESSAGES).                                           *00001610
C*    MAXPMT=1 - USER WANTS MAXIMUM PROMPTING (WITH EXPLANATORY    *00001620
C*              MESSAGES).                                           *00001630
C*  2- PRMT.                                                         *00001640
C*    PRMT=0 - TERMINAL PRINTING ONLY (DEFAULT).                   *00001650
C*      =1 - OFF-LINE PRINTING ONLY.                                 *00001660
C*      =2 - TERMINAL AND OFF-LINE PRINTING.                         *00001670
C*  3- REREAD.                                                       *00001680
C*    REREAD=0 - (REREAD IS ONLY APPLICABLE IF ITER>1) USE THE     *00001690
C*                VARIABLE VALUES OF THE LAST RUN AS DEFAULT VALUES
C*                FOR THE CURRENT RUN (DEFAULT).                   *00001700
C*      =1 - ONCE AGAIN CALL INIT AND THE FIVE READ SUBROUTINES *00001720
C*            TO USE THE VALUES FROM THE INPUT FILES AS DEFAULT   *00001730
C*            VARIABLE VALUES.                                     *00001740
C*****00001750
C                                00001760
C      EXIT = 0                                                       00001770
C      CALL PRMPT1                                                    00001780
C                                00001790
C*****00001800
C*                                TITLE, INITIAL, AND READ SUBROUTINES *00001810
C*  IF USER REQUESTED OFF-LINE OUTPUT, PRINT TITLE PAGE (CALL TITLE). *00001820
C*  IF THIS IS THE FIRST PASS THROUGH THE LCC LOOP (ITER=1) OR IF THE *00001830
C*  USER SET REREAD=1 IN PRMPT1, THEN CALL SUBROUTINE INITIAL      *00001840
C*  TO INITIALIZE CERTAIN VARIABLES TO INITIAL VALUES AND CALL THE *00001850
C*  FIVE READ SUBROUTINES TO INITIALIZE VARIABLE VALUES TO THE VALUES *00001860
C*  FOUND IN THE FILES. EACH READ SUBROUTINE READS A SINGLE INPUT FILE *00001870
C*  AND CHECKS THAT FILE FOR ERRORS. IERROR COUNTS THE NUMBER OF    *00001880
C*  ERRORS FOUND IN THE FIVE FILES. IF IERROR>0 AFTER THE FIVE READ *00001890

```

```

C* SUBROUTINES, THE PROGRAM ISSUES A MESSAGE SAYING IT WILL SOON STOP.*00001900
C*****00001910
C
IF (PRNT.NE.0) CALL TITLE                                00001920
IF (ITER.GT.1.AND.REREAD.EQ.0) GO TO 4                    00001930
CALL INITAL                                              00001940
IERROR = 0                                              00001950
REWIND 11                                              00001960
REWIND 12                                              00001970
REWIND 13                                              00001980
REWIND 14                                              00001990
REWIND 15                                              00002000
CALL READ1                                              00002010
CALL READ2                                              00002020
CALL READ3                                              00002030
CALL READ4                                              00002040
CALL READ5                                              00002050
IF (IERROR.NE.0) WRITE(6,1)                             00002060
IERRI = IERROR                                          00002070
4 CONTINUE                                              00002080
C
C*****00002090
C* PRMPT2                                              00002100
C* CALL THE SECOND OF THE PROMPTING ROUTINES TO DETERMINE: 00002110
C* 1- NAMELIST /GO/, A NAMELIST THAT CONTAINS ALL VARIABLES THAT 00002120
C* WERE READ IN FROM THE FIVE INPUT FILES. BY ASSIGNING VALUES 00002130
C* IN NAMELIST /GO/, THE USER OVERWRITES THE VALUES READ IN FROM 00002140
C* THE INPUT FILES. HENCE, THE USER DOES NOT HAVE TO HAVE 00002150
C* DIFFERENT DATA SETS FOR EVERY LCC RUN: HE CAN USE THE INPUT 00002160
C* FILES AS HIS 'STANDARD' DATA, AND CAN MODIFY WHATEVER 00002170
C* VARIABLE VALUES HE WANTS IN NAMELIST /GO/. 00002180
C* 2- NAMELIST /SENS/, WHICH CONTAINS THE SENSITIVITY ANALYSIS 00002190
C* PARAMETERS. FOR EACH OPTIONAL SENSITIVITY FACTOR KXX (TWO 00002200
C* SENSITIVITY FACTORS, XUC AND K, ARE NOT OPTIONAL), THE 00002210
C* USER CAN ASSIGN VARIABLE LDXXX IN NAMELIST /SENS/ TO THE 00002220
C* NUMBER OF THE LARGEST DERIVATIVES OF TYPE XXX WHICH HE WISHES 00002230
C* TO SEE IN HIS TERMINAL OUTPUT (DEFAULT LDXXX=0). THE USER CAN 00002240
C* ALSO 00002250
C* ASSIGN A VALUE TO PINC, THE FRACTIONAL INCREASE IN FACTORS 00002260
C* WHICH THE PROGRAM SHOULD USE IN THE SENSITIVITY ANALYSIS 00002270
C* LDERV (DEFAULT OF 12) TELLS THE NUMBER (UNLESS A PARTICULAR 00002280
C* LDXXX EXCEEDS LDERV) OF DERIVATIVES OF ALL TYPES WHICH THE 00002290
C* USER WANTS TO SEE IN HIS OFF-LINE OUTPUT. 00002300
C* IF EXIT=1 AFTER PRMPT2, THE USER WANTS TO EXIT FROM THE PROGRAM. 00002310
C* AFTER PRMPT2, ALL INPUT INTO THE PROGRAM HAS BEEN DETERMINED. 00002320
C* THE CALCULATION OF THE LCC WILL FOLLOW. 00002330
C* AFTER PRMPT2 LDERV IS SET TO THE MINIMUM OF ITS VALUE AFTER 00002340
C* /SENS/ IS READ AND N (LARGEST LRU INDEX NUMBER). 00002350
C*****00002360
C* CALL PRMPT2 00002370
C* IF (EXIT.EQ.1) GO TO 888 00002380
C* LDERV = MIN0(LDERV,N) 00002390
C
C*****00002400
C* ERRCHK 00002410
C* ERRCHK CONTAINS MOST OF THE CHECKS ON THE INPUT DATA. THERE ARE 00002420
C* TWO CLASSES OF ERROR CHECKS: FATAL ONES THAT INCREMENT IERROR AND 00002430
C* NON-FATAL ONES THAT INCREMENT IWARN. SOME FATAL CHECKS HAVE 00002440
C* ALREADY BEEN DONE IN THE READ SUBROUTINES; ERRCHK DOES FURTHER 00002450
C* CHECKS ON RL,SRTS1+CRS1+WOR1+DRTS1,CRS2+WOR2+DRTS2,WOR3+DRTS3, 00002460
C* NIBI,SMH+RMH+BCMH: NON-FATAL CHECKS BY ERRCHK INCLUDE: 00002470
C* SMTBI,SRCT,SOSI,SMH+RMH+BCMH,CONF,CPTI,NSP 00002480
00002490
00002500
00002510

```

```

C* IF A VARIABLE FAILS A FATAL CHECK, AN ERROR MESSAGE IS PRINTED AND *00002520
C* THE LCC WILL BE SKIPPED. IF A VARIABLE FAILS A NON-FATAL *00002530
C* CHECK, A WARNING MESSAGE IS ISSUED AND A DEFAULT VALUE IS ASSIGNED.*00002540
C*****00002550
C 00002560
      IWARN = 0 00002570
      IERROR = IERRI 00002575
      CALL ERRCHK 00002580
      IF (IWARN.EQ.0) GO TO 5 00002590
      WRITE(6,2) IWARN 00002600
      IF (PRNT.NE.0) WRITE(7,2) IWARN 00002610
5 IF (IERROR.EQ.0) GO TO 6 00002620
      WRITE(6,3) IERROR 00002630
      IF (PRNT.NE.0) WRITE(7,3) IERROR 00002640
      GO TO 888 00002650
6 CONTINUE 00002660
C 00002670
C*****00002680
C* PRELIMINARY CALCULATIONS *00002690
C* CALL THE FIVE SUBROUTINES THAT CALCULATE THE AUXILIARY VARIABLES. *00002700
C*****00002710
C 00002720
      CALL AUXIL1 00002730
      CALL AUXIL2 00002740
      CALL AUXIL3 00002750
      CALL AUXIL4 00002760
      CALL AUXIL5 00002770
C 00002780
C*****00002790
C* SUM UP THE COST ELEMENTS. *00002800
C*****00002810
C 00002820
      CALL COST1 00002830
      CALL COST2 00002840
      CALL COST3 00002850
      CALL COST4 00002860
      CALL COST5 00002870
      CALL COST6 00002880
      CALL COST7 00002890
      CALL COST8 00002900
      CALL COST9 00002910
      CALL COST10 00002920
C 00002930
C*****00002940
C* SUBSEQUENT CALCULATIONS *00002950
C* CALCULATE THE NON-COST VARIABLES AND THE DESIRED DERIVATIVES. *00002960
C* THREE DERIVATIVES ARE ALWAYS CALCULATED (TDFR(I),TDXUC,TDK), BUT *00002970
C* ONLY TWO DERIVATIVES ARE ALWAYS PRINTED (TDXUC,TDK: TDFR IS ALWAYS *00002980
C* CALCULATED BECAUSE IT IS USED TO CALCULATE TDK). FOR ANY OTHER *00002990
C* DERIVATIVE (CALL IT TD---), THE SUBROUTINE (D---) THAT CALCULATES *00003000
C* THE DERIVATIVE IS ONLY CALLED IF THE USER REQUESTS OFF-LINE PRINT- *00003010
C* ING OF A NON-ZERO NUMBER (LDERV) OF LARGEST DERIVATIVES OF ANY TYPE*00003020
C* OR A NON-ZERO NUMBER OF THE LARGEST DERIVATIVES OF THIS TYPE (LD---)*00003030
C* TELLS THE NUMBER OF LARGEST TD--- VALUES THAT THE USER WANTS *00003040
C* OF THIS TYPE PRINTED AT THE TERMINAL) AT THE TERMINAL. *00003050
C* THE SUBROUTINE (D---) WILL CALCULATE THE DERIVATIVE *00003060
C* (TD---(I)) FOR EACH LRU I AND THEN WILL CALL SUBROUTINE TDSORT TO *00003070
C* BUBBLE INTO THE FIRST NN (NN=MAX(LDERV,LD---)) POSITIONS OF ARRAY *00003080
C* (ID---) THE INDEX NOS. OF THE LRUS WITH THE LARGEST TD--- VALUES. *00003090
C*****00003100
C 00003110
      CALL NONCOS 00003120
      CALL DXUC 00003130

```



```

CALL DFR                                00003140
CALL DK                                00003150
LTEMP = 1                              00003160
IF (LDERV.EQ.0.OR.PRNT.EQ.0) LTEMP=0  00003170
IF (LDUC.GT.0.OR.LTEMP.GT.0) CALL DUC  00003180
IF (LDWOR.GT.0.OR.LTEMP.GT.0) CALL DWOR 00003190
IF (LDPPR.GT.0.OR.LTEMP.GT.0) CALL DPPR 00003200
IF (LDCMH.GT.0.OR.LTEMP.GT.0) CALL DCMH 00003210
IF (LDDMH.GT.0.OR.LTEMP.GT.0) CALL DDMH 00003220
IF (LDRM.GT.0.OR.LTEMP.GT.0) CALL DRM   00003230
IF (LDSRTS.GT.0.OR.LTEMP.GT.0) CALL DSRTS 00003240
IF (LDCRTS.GT.0.OR.LTEMP.GT.0) CALL DCRTS 00003250
IF (LDDRTS.GT.0.OR.LTEMP.GT.0) CALL DDRTS 00003260
IF (LDMCI.GT.0.OR.LTEMP.GT.0) CALL DMCI 00003270
C                                     00003280
C*****                                00003290
C*                                     PRMPT3 *00003300
C* CALL THE THIRD OF THE PROMPTING ROUTINES TO: *00003310
C* 1- TELL THE USER THAT THE LCC CALCULATIONS HAVE BEEN COMPLETED. *00003320
C* 2- GIVE THE USER A CHANCE TO EXIT FROM THE PROGRAM. *00003330
C* 3- ALLOW THE USER A CHANCE TO ADJUST TERMINAL TO A NEW PAGE *00003340
C* BEFORE OTAB1 WRITES AT HIS TERMINAL. *00003350
C*****                                00003360
C                                     00003370
C*                                     PRMPT3 *00003380
C* IF (EXIT.EQ.1) GO TO 888 *00003390
C                                     00003400
C*****                                00003410
C*                                     INPUT TABLES *00003420
C* PRINT THE INPUT TABLES IF THE USER REQUESTS HARD COPY. *00003430
C*****                                00003440
C                                     00003450
C* IF (PRNT.EQ.0) GO TO 7 *00003460
C* CALL ITAB1 *00003470
C* CALL ITAB2 *00003480
C* CALL ITAB3 *00003490
C* CALL ITAB4 *00003500
C* CALL ITAB5 *00003510
C* 7 CONTINUE *00003520
C                                     00003530
C*****                                00003540
C* PRINT THE OUTPUT TABLES. *00003550
C* OUTPUT TABLE 1 AND THE SENSITIVITY ANALYSIS TABLE ARE ALWAYS *00003560
C* PRINTED SOMEWHERE. THE OTHER OUTPUT TABLES ARE ONLY PRINTED (ONLY *00003570
C* OFF-LINE) IF THE USER REQUESTS OFF-LINE PRINTING. *00003580
C* IF USER WANTS TERMINAL PRINTING, CALL PRMPT4 TO ALLOW USER TO *00003590
C* ADJUST TERMINAL TO NEW PAGE BEFORE OSENS IS CALLED. *00003600
C*****                                00003610
C                                     00003620
C* CALL OTAB1 *00003630
C* IF (PRNT.EQ.0) GO TO 8 *00003640
C* CALL OTAB2 *00003650
C* CALL OTAB3 *00003660
C* CALL OTAB4 *00003670
C* CALL OTAB5 *00003680
C* 8 CONTINUE *00003690
C* IF (PRNT.NE.1) CALL PRMPT4 *00003700
C* CALL OSENS *00003710
C                                     00003720
C*****                                00003730
C* PRMPT5 *00003740
C* CALL THE FIFTH AND FINAL PROMPTING ROUTINE TO ASK IF ANOTHER RUN *00003750
C* IS DESIRED. IF NOT, STOP. *00003760

```


C*****	00003770
C	00003780
888 CALL PRMPT5	00003790
IF (REDO.EQ.0) STOP	00003800
999 CONTINUE	00003810
C	00003820
STOP	00003830
END	00003840

C	SUBROUTINE TITLE	00000010
C	*****	00000020
C	*****	00000030
C	* SUBROUTINE TO PRINT TITLE PAGE FOR OFF-LINE OUTPUT.	*00000040
C	*****	00000050
C		00000060
C	COMMON /TITL/ TITL(28)	00000070
C		00000080
1000	FORMAT (1H1//////////)	00000090
+	35X,39H*****//35X,1H*,37X,1H*/	00000100
+	35X,39H* SEEK IGLOO LIFE CYCLE COST MODEL */35X,1H*,37X,1H*/	00000110
+	35X,39H* FIVE INPUT TABLES */35X,1H*,37X,1H*/	00000120
+	35X,39H* SIX OUTPUT TABLES */35X,1H*,37X,1H*)	00000130
1100	FORMAT (35X,9H* RUN: ,27A1,3H */35X,1H*,37X,1H*/	00000140
+	35X,39H*****//)	00000150
C		00000160
C		00000170
	WRITE(7,1000)	00000180
	WRITE(7,1100) (TITL(L),L=1,27)	00000190
C		00000200
C		00000210
	RETURN	00000220
	END	00000230

	SUBROUTINE SNSSET	00000010
C		00000020
C*****		00000030
C*	THIS SUBROUTINE INITIALIZES SENSITIVITY ANALYSIS VARIABLES TO	*00000040
C*	DEFAULT VALUES.	*00000050
C*****		00000060
C		00000070
	COMMON /SENS/ FINC,LDCMH,LDCRTS,LDDMH,LDDRTS,LDERV,LDFPR,LDFR,	00000080
	+ LDMCI,LDRM,LDSRTS,LDWOR,LDUC	00000090
C		00000100
	FINC = 0.1	00000110
	LDCMH = 0	00000120
	LDCRTS = 0	00000130
	LDDMH = 0	00000140
	LDDRTS = 0	00000150
	LDERV = 12	00000160
	LDFPR = 0	00000170
	LDFR = 0	00000180
	LDMCI = 0	00000190
	LDRM = 0	00000200
	LDSRTS = 0	00000210
	LDWOR = 0	00000220
	LDUC = 0	00000230
C		00000240
	RETURN	00000250
	END	00000260

```

SUBROUTINE INITAL                                00000010
C                                                    00000020
C*****00000030
C* THIS SUBROUTINE INITIALIZES CERTAIN VARIABLES TO CERTAIN DESIRED *00000040
C* VALUES. SCALARS ARE INITIALIZED FIRST, AND THEN ARRAYS. *00000050
C*****00000060
C                                                    00000070
COMMON /INIT/ CONFLO,MAXFA,MAXLR,MAXSE          00000080
C                                                    00000090
COMMON /RD1/ ADCM,ADPH,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,CONF,CPHI, 00000100
+ CPMP,CPMT,CRCT,DAA,DLR,DRCT,DS,              00000110
+ PAC(10),PCS,PSD,H,INC,K,H,MCRS,MXHS,          00000120
+ NCP,NDP,NSP,OST,PIUP,PHE,PPRS,PRODX,REFURB,RMC, 00000130
+ SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC,         00000140
+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR 00000150
INTEGER CADRE                                00000160
REAL INC,K,MCRS,MXHS                        00000170
C                                                    00000180
COMMON /RD2/ A,COC(150),COD(150),COS(150),INOS(150),NSEC(150), 00000190
+ NSED(150),NSES(150),SEC(150),SENAME(150,24) 00000200
REAL NSEC,NSED,NSES                        00000210
INTEGER A                                00000220
C                                                    00000230
COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),MCI(200), 00000240
+ MTBI(200),N,PA(200),QPA(200),QR(200),RH(200), 00000250
+ UC(200)                                     00000260
INTEGER QPA,QR                             00000270
REAL MTBI                                00000280
C                                                    00000290
COMMON /RD5/ BCMH(200),CBCMH(200),CMH(200),CRTS1(200),CRTS2(200), 00000300
+ DBCMH(200),DMH(200),DRTS1(200),DRTS2(200), 00000310
+ DRTS3(200),FPR(200),IMH(200),INO15(200),N15, 00000320
+ PAMH(200),RIP(200),RL(200),RHH(200),SMH(200),SRTS1(200), 00000330
+ WOR1(200),WOR2(200),WOR3(200)             00000340
REAL IMH                                00000350
INTEGER RL                                00000360
C                                                    00000370
MXHS = 1300.                                00000380
MAXFA = 10                                  00000390
MAXLR = 200                                 00000400
MAXSE = 150                                 00000410
CONFLO = 0.84134                            00000420
C                                                    00000430
DO 1 L=1,10                                  00000440
PAC(L) = 0.                                  00000450
1 CONTINUE                                    00000460
C                                                    00000470
DO 2 J=1,MAXSE                                00000480
NSES(J) = 0.                                  00000490
NSEC(J) = 0.                                  00000500
NSED(J) = 0.                                  00000510
2 CONTINUE                                    00000520
C                                                    00000530
DO 3 I=1,MAXLR                                00000540
UC(I) = 0.                                    00000550
FPR(I) = 0.                                    00000560
CMH(I) = 0.                                    00000570
DMH(I) = 0.                                    00000580
RH(I) = 0.                                    00000590
MCI(I) = 0.                                    00000600
QPA(I) = 0.                                    00000610
MTBI(I) = 990000000.                          00000620
3 CONTINUE                                    00000630

```


C

RETURN
END

00000640
00000650
00000660

```

SUBROUTINE PRMPT1
C
C*****
C* SUBROUTINE PRMPT1
C* FIRST OF FOUR PROMPTING SUBROUTINES TO READ IN USER INPUTS FROM
C* THE TERMINAL. IF THIS IS THE FIRST CALL OF THE LCC:
C* 1- PRINT TITLE ('SEEK IGLOO LIFE CYCLE COST MODEL,ETC')
C* 2- ASK USER FOR MIN OR MAX PROMPTING (MAX=LONG PROMPT COMMENTS)
C* 3- ASK USER WHERE HE WANTS HIS OUTPUT. (PRNT=0 TERMINAL ONLY;
C* PRNT=1 OFF-LINE ONLY; PRNT=2 BOTH PLACES.)
C* 4- IF USER REQUEST OFF-LINE OUTPUT, GET A NAME FOR THE RUN.
C* IF THIS IS A SUBSEQUENT CALL FOR THE LCC:
C* 1- NOTIFY USER THAT VARIABLES ARE AS THEY WERE AFTER LAST
C* NAMELISTS WERE SUBMITTED.
C* 2- ASK USER IF HE WANTS TO REREAD INPUT FILES (REREAD=1).
C* 3- ASK MIN OR MAX PROMPTING ONLY IF LAST RUN WAS MAX PROMPTING.
C* 4- ASK USER WHERE HE WANTS HIS OUTPUT. (PRNT=0 TERMINAL ONLY;
C* PRNT=1 OFF-LINE ONLY; PRNT=2 BOTH PLACES)
C* 5- IF USER REQUEST OFF-LINE OUTPUT, GET A NAME FOR THE RUN.
C*****
COMMON /CNTL/ EXIT,ITER,MAXPMT,PRNT,REDO,REREAD
INTEGER EXIT,PRNT,REDO,REREAD

COMMON /TITL/ TITL(28)

DATA BK/1H /,CHM/1HM/,CHI/1HI/,CHN/1HN/,CHA/1HA/,CHX/1HX/,
+ CHY/1HY/

1 FORMAT (1X/47H SEEK IGLOO LIFE CYCLE COST MODEL - VERSION 1.0)
2 FORMAT (1X/50H AT THIS POINT, VARIABLE VALUES ARE AS THEY WERE A,
+ 13HFTER THE LAST/
+ 53H NAMELISTS WERE SUBMITTED. DO YOU WISH TO RESET NAME,
+ 9HLIST /GO//
+ 54H VARIABLES TO THE VALUES FOUND IN THE INPUT FILES (Y O,
+ 6HR N)-?)
3 FORMAT (1X/44H MINIMUM OR MAXIMUM PROMPTING (MIN OR MAX)-?)
4 FORMAT (3A1)
5 FORMAT (1X/50H SUBMIT 'MIN' OR 'MAX' STARTING IN COLUMN 1. NOTH,
+ 15HING ELSE WORKS.)
6 FORMAT (30H OUTPUT AT TERMINAL (Y OR N)-?)
7 FORMAT (A1)
8 FORMAT (49H SUBMIT 'Y' OR 'N' STARTING IN COLUMN 1. NOTHING,
+ 12H ELSE WORKS.)
9 FORMAT (38H OUTPUT ON OFF-LINE PRINTER (Y OR N)-?)
10 FORMAT (29H SUBMIT A TITLE FOR THIS RUN:)
11 FORMAT (28A1)
12 FORMAT (/51H SET EXIT=1 IN EITHER NAMELIST IF YOU WANT TO EXIT.)

C
C
IF (ITER.NE.1) GO TO 16
WRITE(6,1)
GO TO 18
16 CONTINUE
WRITE(6,2)
REREAD = 2
STR1 = BK
17 READ(5,7) STR1
IF (STR1.EQ.CHY) REREAD = 1
IF (STR1.EQ.CHN) REREAD = 0
IF (REREAD.NE.2) GO TO 18
WRITE(6,8)
GO TO 17

```

```

18 CONTINUE
   IF (ITER.GT.1.AND.MAXPMT.NE.1) GO TO 20
   WRITE(6,3)
   MAXPMT = 2
19 READ(5,4) STR1,STR2,STR3
   IF (STR1.EQ.CHN.AND.STR2.EQ.CHI.AND.STR3.EQ.CHN) MAXPMT = 0
   IF (STR1.EQ.CHN.AND.STR2.EQ.CHA.AND.STR3.EQ.CHX) MAXPMT = 1
   IF (MAXPMT.NE.2) GO TO 20
   WRITE(6,5)
   GO TO 19
20 CONTINUE
   MM1 = 2
   WRITE(6,6)
21 READ(5,7) STR1
   IF (STR1.EQ.CHY) MM1 = 1
   IF (STR1.EQ.CHN) MM1 = 0
   IF (MM1.NE.2) GO TO 22
   WRITE(6,8)
   GO TO 21
22 MM2 = 2
   STR1 = BK
   WRITE(6,9)
23 READ(5,7) STR1
   IF (STR1.EQ.CHY) MM2 = 1
   IF (STR1.EQ.CHN) MM2 = 0
   IF (MM2.NE.2) GO TO 24
   WRITE(6,8)
   GO TO 23
24 CONTINUE
   IF (MM2.EQ.0) PRNT=0
   IF (MM1.EQ.0.AND.MM2.EQ.1) PRNT = 1
   IF (MM1.EQ.1.AND.MM2.EQ.1) PRNT = 2
   IF (PRNT.EQ.0) GO TO 88
   WRITE(6,10)
   READ(5,11) (TITL(L),L=1,28)
88 IF (MAXPMT.EQ.1.OR.ITER.EQ.1) WRITE(6,12)
C
   RETURN
   END
   SUBROUTINE PRMPT2
C
C*****
C* SUBROUTINE PRMPT2
C* SECOND OF THE PROMPTING ROUTINES. PROMPTS THE USER FOR NAMELISTS
C* /GO/ (WHICH CONTAINS ALL VARIABLES READ IN FROM THE FIVE INPUT
C* FILES AND ALLOWS THE USER TO OVERRIDE THOSE VALUES IN REAL TIME)
C* AND /SENS/ (WHICH CONTAINS PARAMETERS FOR THE SENSITIVITY
C* ANALYSIS: FOR EACH OPTIONAL DERIVATIVE TD---, LD--- TELLS HOW MANY
C* OF THE LARGEST TD--- DERIVATIVES THE USER WANTS TO SEE IN HIS
C* TERMINAL OUTPUT. LDERV TELLS HOW MANY OF THE LARGEST DERIVATIVES
C* OF ALL TYPES THE USER WANTS TO SEE IN HIS OFF-LINE OUTPUT.
C*****
C
COMMON /CNTL/ EXIT,ITER,MAXPMT,PRNT,REDO,REREAD
INTEGER EXIT,PRNT,REDO,REREAD
C
COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCHP,CDR,CDWH,CLR,CONF,CPHI,
+ CPMP,CPWT,CRCT,DAA,DLN,DPCT,DS,
+ PAC(10),PCS,PSD,H,INC,K,H,MCRS,NKHS,
+ NCP,NDP,NSP,OST,PIUP,PHE,PPRS,PRCDX,REFURB,RMC,
+ SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC,
+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR
INTEGER CADRE

```

	REAL INC,K,MCRS,MXHS	00001270
C	COMMON /RD2/ A,COC(150),COD(150),COS(150),INOS(150),NSEC(150),	00001280
	+ NSED(150),NSES(150),SEC(150),SENAME(150,24)	00001290
	REAL NSEC,NSED,NSES	00001300
	INTEGER A	00001310
C	COMMON /RD3/ ENYR(10),PA,PANAME(10,24),HWCF(10),INOPA(10),MMH(10),	00001320
	+ SIZE(10),SMI(10),SWCF(10),SWPIX(10),SWVAR(10),	00001330
	+ WEIGHT(10)	00001340
	INTEGER PA	00001350
	REAL MMH	00001360
C	COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200),	00001370
	+ MTBI(200),N,PA(200),QPA(200),QR(200),RM(200),	00001380
	+ UC(200)	00001390
	INTEGER QPA,QR	00001400
	REAL MTBI	00001410
C	COMMON /RD5/ BCMH(200),CBCMH(200),CMH(200),CRTS1(200),CRTS2(200),	00001420
	+ DBCMH(200),DMH(200),DRTS1(200),DRTS2(200),	00001430
	+ DRTS3(200),FPR(200),INH(200),INO15(200),N15,	00001440
	+ PANH(200),RIP(200),RL(200),RMH(200),SMH(200),SRTS1(200),	00001450
	+ WOR1(200),WOR2(200),WOR3(200)	00001460
	REAL INH	00001470
	INTEGER RL	00001480
C	COMMON /SENS/ FINC,LDCMH,LDCRTS,LDDMH,LDDRTS,LDERV,LDFPR,LDFR,	00001490
	+ LDMCI,LDRM,LDSRTS,LDWOR,LDUC	00001500
C	NAMLIST/GO/ ADCH,ADPH,B,CAA,CADRE,CCHP,CDR,CDWH,CLR,CONF,CPHI,	00001510
	+ CPMP,CPMT,CRCT,DAA,DLR,DRCT,DS,	00001520
	+ PAC,FCS,PSD,H,INC,K,M,MCRS,MXHS,	00001530
	+ MCP,NDP,NSP,OST,PIUP,PHE,PPRS,PRODX,REFURB,RMC,	00001540
	+ SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC,	00001550
	+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR,	00001560
	+ A,COC,COD,COS,INOS,NSEC,	00001570
	+ NSED,NSES,SEC,SENAME,	00001580
	+ ENYR,PA,PANAME,HWCF,INOPA,MMH,	00001590
	+ SIZE,SMI,SWCF,SWPIX,SWVAR,	00001600
	+ WEIGHT,	00001610
	+ INO,LRNAME,LRPART,HCI,	00001620
	+ MTBI,N,PA,QPA,QR,RM,	00001630
	+ UC,	00001640
	+ BCMH,CBCMH,CMH,CRTS1,CRTS2,	00001650
	+ DBCMH,DMH,DRTS1,DRTS2,	00001660
	+ DRTS3,FPR,INH,INO15,N15,	00001670
	+ PANH,RIP,RL,RMH,SMH,SRTS1,	00001680
	+ WOR1,WOR2,WOR3,EXIT	00001690
C	NAMLIST/SENS/ FINC,LDCMH,LDCRTS,LDDMH,LDDRTS,LDERV,LDFPR,LDFR,	00001700
	+ LDMCI,LDRM,LDSRTS,LDWOR,LDUC,EXIT	00001710
C	1 FORMAT (1X)	00001720
	2 FORMAT (51H NAMLIST /GO/ CONTAINS ALL VARIABLES FOUND IN THE,	00001730
	+ 13H INPUT FILES.)	00001740
	3 FORMAT (52H AT THIS POINT, NAMLIST /GO/ VARIABLES CONTAIN VAL,	00001750
	+ 13HUES AS IN THE/	00001760
	+ 18H THE INPUT FILES.)	00001770
	4 FORMAT (52H AT THIS POINT, NAMLIST /GO/ VARIABLES ARE AS THEY,	00001780
	+ 11H WERE AFTER/	00001790
	+ 39H THE LAST NAMLIST /GO/ WAS SUBMITTED.)	00001800
	5 FORMAT (53H TO USE THESE VALUES, SUBMIT AN EMPTY NAMLIST /GO./	00001810
		00001820
		00001830
		00001840
		00001850
		00001860
		00001870
		00001880
		00001890


```

+      52H TO OVERRIDE ANY OF THESE VALUES, SUBMIT AN NON-EMP, 00001900
+      17HTY NAMELIST /GO/.) 00001910
6  FORMAT (41H SUBMIT NAMELIST /GO/ IN NAMELIST FORMAT:) 00001920
7  FORMAT (53H NAMELIST /SENS/ CONTAINS VARIABLES THAT CONTROL THE/ 00001930
+      53H DISPLAY OF THE SENSITIVITY ANALYSIS. ) 00001940
8  FORMAT (53H AT THIS POINT, THE TERMINAL DISPLAYS SENSITIVITY WI, 00001950
+      18HTH RESPECT TO ONLY/32H GLOBAL FAILURE RATE (K FACTOR), 00001960
+      40H AND GLOBAL EQUIPMENT COST (XUC FACTOR).) 00001970
9  FORMAT (52H AT THIS POINT, THE TERMINAL SENSITIVITY DISPLAY IS/ 00001980
+      32H AS IT WAS ON THE PREVIOUS RUN.) 00001990
10 FORMAT (53H FOR THE SAME TERMINAL DISPLAY, SUBMIT AN EMPTY NAME, 00002000
+      12HLIST /SENS/./ 00002010
+      54H FOR A DIFFERENT TERMINAL DISPLAY, SUBMIT A NON-EMPTY, 00002020
+      17H NAMELIST /SENS/.) 00002030
11 FORMAT (43H SUBMIT NAMELIST /SENS/ IN NAMELIST FORMAT:) 00002040
C      00002050
      WRITE(6,1) 00002060
      IF (MAXPMT.NE.1) GO TO 30 00002070
      WRITE(6,2) 00002080
      IF (ITER.EQ.1.OR.REREAD.EQ.1) WRITE(6,3) 00002090
      IF (ITER.GT.1.AND.REREAD.NE.1) WRITE(6,4) 00002100
      WRITE(6,5) 00002110
30  WRITE(6,6) 00002120
      READ(5,GO) 00002130
      IF (EXIT.EQ.1) RETURN 00002140
      WRITE(6,1) 00002150
      IF (MAXPMT.NE.1) GO TO 31 00002160
      LDTOT = LDCMH + LDCRTS + LDDMH + LDDRTS + LDFPR + LDPR + LDRM 00002170
+      + LDMCI + LDSRTS + LDWOR + LDUC 00002180
      WRITE(6,7) 00002190
      IF (LDTOT.EQ.0) WRITE(6,8) 00002200
      IF (LDTOT.NE.0) WRITE(6,9) 00002210
      WRITE(6,10) 00002220
31  WRITE(6,11) 00002230
      READ(5,SENS) 00002240
C      00002250
      RETURN 00002260
      END 00002270
      SUBROUTINE PRMPT3 00002280
C      00002290
C*****00002300
C*      SUBROUTINE PRMPT3 *00002310
C*  THIRD OF PROMPTING ROUTINES THAT TELLS USER THAT LCC HAS BEEN *00002320
C*  COMPLETED AND THAT GIVES THE USER A CHANCE TO EXIT. *00002330
C*****00002340
C      00002350
      COMMON /CNTL/ EXIT,ITER,MAXPMT,PRNT,REDO,REREAD 00002360
      INTEGER EXIT,PRNT,REDO,REREAD 00002370
C      00002380
      DATA ECHAR/1HE/,BK/1H / 00002390
C      00002400
2  FORMAT (1X/15H LCC COMPLETED.) 00002410
3  FORMAT (49H IF YOU WISH TO EXIT, HIT -E-, THEN HIT -RETURN-;, 00002420
+      12H OTHERWISE,) 00002430
4  FORMAT (46H ADJUST TERMINAL TO NEW PAGE AND HIT -RETURN-.) 00002440
5  FORMAT (A1) 00002450
C      00002460
C      00002470
      STR = BK 00002480
      WRITE(6,2) 00002490
      IF (MAXPMT.EQ.1) WRITE(6,3) 00002500
      WRITE(6,4) 00002510
      READ(5,5) STR 00002520

```

```

      IF (STR.EQ.ECHAR) EXIT=1
C
      RETURN
      END
      SUBROUTINE PRMPT4
C
C*****
C* SUBROUTINE PRMPT4
C* THIRD OF PROMPTING ROUTINES THAT ASKS USER TO ADJUST PAPER TO NEW
C* PAGE FOR OSENS AND GIVES THE USER A CHANCE TO EXIT.
C*****
      COMMON /CNTL/ EXIT,ITER,MAXPMT,PRNT,REDO,REREAD
      INTEGER EXIT,PRNT,REDO,REREAD
C
      DATA ECHAR/1HE/,BK/1H /
C
      2  FORMAT (1X)
      3  FORMAT (49H IF YOU WISH TO EXIT, HIT -E-, THEN HIT -RETURN-;,
      *    12H OTHERWISE,)
      4  FORMAT (46H ADJUST TERMINAL TO NEW PAGE AND HIT -RETURN-. )
      5  FORMAT (A1)
C
      STR = BK
      WRITE (6,2)
      IF (MAXPMT.EQ.1) WRITE (6,3)
      WRITE (6,4)
      READ (5,5) STR
      IF (STR.EQ.ECHAR) EXIT=1
C
      RETURN
      END
      SUBROUTINE PRMPT5
C
C*****
C* SUBROUTINE PRMPT5
C* FIFTH AND FINAL PROMPTING ROUTINE TO ASK IF ANOTHER RUN IS
C* DESIRED. IF SO, REDO = 1. IF NOT, REDO = 0.
C*****
      COMMON /CNTL/ EXIT,ITER,MAXPMT,PRNT,REDO,REREAD
      INTEGER EXIT,PRNT,REDO,REREAD
C
      DATA YES/1HY/,CNO/1HN/
C
      1  FORMAT (1X/23H ANOTHER RUN (Y OR N) -?)
      2  FORMAT (A1)
      3  FORMAT (49H SUBMIT 'Y' OR 'N' STARTING IN COLUMN 1. NOTHING,
      *    12H ELSE WORKS.)
C
      REDO = 2
      WRITE (6,1)
      4  READ (5,2) STR
      IF (STR.EQ.YES) REDO=1
      IF (STR.EQ.CNO) REDO=0
      IF (REDO.NE.2) GO TO 5
      WRITE (6,3)
      GO TO 4
      5  CONTINUE
C
      RETURN
      END

```

```

SUBROUTINE READ1
C
C*****
C* SUBROUTINE READS NAMELIST/MISC/FROM THE UNIT 11 FILE: MISCELLANEOUS
C* INPUTS (MAINLY SCALARS).
C*****
C
COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCHP,CDR,CDWH,CLR,CONF,CPhi,
+ CPNP,CPWT,CRCT,DAA,DLR,DRCT,DS,
+ PAC(10),PCS,PSD,H,INC,K,M,MCRS,MXHS,
+ NCP,NDP,NSP,OST,PIUP,PHE,PPRS,PRODX,REFURB,RMC,
+ SA,SAA,SHTBI,SOSI,SPRS,SRCT,SW,SWFAC,
+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR
INTEGER CADRE
REAL INC,K,MCRS,MXHS
C
NAMELIST /MISC/ ADCH,ADPH,B,CAA,CADRE,CCHP,CDR,CDWH,CLR,CONF,CPhi,
+ CPNP,CPWT,CRCT,DAA,DLR,DRCT,DS,
+ PAC,PCS,PSD,H,INC,K,M,MCRS,MXHS,
+ NCP,NDP,NSP,OST,PIUP,PHE,PPRS,PRODX,REFURB,RMC,
+ SA,SAA,SHTBI,SOSI,SPRS,SRCT,SW,SWFAC,
+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR
READ(11,MISC)
RETURN
END
SUBROUTINE ITAB1
C
C*****
C* SUBROUTINE PRINTS A INPUT TABLE 1 WHICH DISPLAYS THE MISCELLANEOUS
C* INPUTS (MAINLY SCALARS).
C*****
C
COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCHP,CDR,CDWH,CLR,CONF,CPhi,
+ CPNP,CPWT,CRCT,DAA,DLR,DRCT,DS,
+ PAC(10),PCS,PSD,H,INC,K,M,MCRS,MXHS,
+ NCP,NDP,NSP,OST,PIUP,PHE,PPRS,PRODX,REFURB,RMC,
+ SA,SAA,SHTBI,SOSI,SPRS,SRCT,SW,SWFAC,
+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR
INTEGER CADRE
REAL INC,K,MCRS,MXHS
C
1000 FORMAT (1H1,30X,43HINPUT TABLE 1: MISCELLANEOUS SCALAR INPUTS)
1100 FORMAT (1H0//7X,7HGENERAL//
+ 15X,22HK - RELIABILITY FACTOR,30X,F12.2)
1200 FORMAT (15X,19HM - NUMBER OF SITES,37X,I5,10H SITES/
+ 12X,32HPIUP - NUMBER OF OPERATING YEARS,23X,F12.2,7H YEARS/
+ 13X,22HXUC - UNIT COST FACTOR,32X,F12.2/
+ 13X,28HYOH - YEARLY OPERATING HOURS,25X,F13.2,7H HOURS)
1300 FORMAT (1H0//7X,27HDEVELOPMENT AND PRODUCTION //
+ 13X,28HPSD - FULL SCALE DEVELOPMENT,26X,
+ F12.3,15H $ IN MILLIONS/
+ 13X,29HPME - PRIME MISSION EQUIPMENT,25X,F12.3,
+ 15H $ IN MILLIONS/
+ 11X,35HPRODX - ASSOCIATED PRODUCTION COSTS,21X,F12.3,
+ 15H $ IN MILLIONS/
+ 10X,22HREFURB - REFURBISHMENT,35X,F12.3,15H $ IN MILLIONS/
+ 14X,25HSW - SOFTWARE DEVELOPMENT,28X,F12.3,15H $ IN MILLIONS/
+ 13X,31HVAL - DESIGN VALIDATION ,23X,F12.3,

```


	+ 15H \$ IN MILLIONS)	00000640
1400	FORMAT (1H0/7X,11HMAINTENANCE//	00000650
	+ 12X,45HADCM - AVERAGE LENGTH OF EMERGENCY CORRECTIVE/	00000660
	+ 19X,16HMAINTENANCE TRIP,32X,F12.2,6H DAYS/	00000670
	+ 12X,46HADPM - DURATION OF PREVENTIVE MAINTENANCE TRIP,	00000680
	+ 9X,F12.2,6H DAYS/	00000690
	+ 13X,28HCAA - AVAILABLE WORK TIME FO,	00000700
	+ 23HR CHF RADAR MAINTENANCE,3X,F12.2,24H HOURS PER YEAR PER MAN/	00000710
	+ 12X,52HCCMP - AVERAGE NUMBER OF CHF PERSONNEL PER EMERGENCY/	00000720
	+ 19X,27HCORRECTIVE MAINTENANCE TRIP,21X,F12.2,5H MEN/	00000730
	+ 13X,32HCDR - TDY RATE FOR CHF PERSONNEL,22X,F12.2,	00000740
	+ 11H \$ PER DAY/	00000750
	+ 12X,49HCDWH - CHF WORKING HOURS ON SITE MAINTENANCE TRIP,	00000760
	+ 6X,F12.2,23H HOURS PER DAY PER MAN/	00000770
	+ 13X,30HCLR - HOURLY LABOR RATE AT CHF,24X,F12.2,12H \$ PER HOUR/	00000780
	+ 12X,51HCPMI - SHORTEST PREVENTIVE MAINTENANCE INTERVAL FOR/	00000790
	+ 19X,19HCHF PERSONNEL TRIPS,29X,F12.2,7H HOURS)	00000800
1500	FORMAT (12X,39HCPMP - AVERAGE NUMBER OF CHF PERSONNEL ,	00000810
	+13HPR SCHEDULED/19X,27HPREVENTIVE MAINTENANCE TRIP,21X,F12.2,2X,	00000820
	+3HEN/13X,52HDAA - AVAILABLE WORK TIME FOR DEPOT REPAIR MAINTENAN,	00000830
	+ 2HCE,F12.2,24H HOURS PER YEAR PER MAN/	00000840
	+ 13X,32HDLR - HOURLY LABOR RATE AT DEPOT,22X,F12.2,	00000850
	+ 12H \$ PER HOUR/	00000860
	+14X,48HDS - FRACTION OF FAILURES AUTOMATICALLY DETECTED,5X,F12.2/	00000870
	+ 15X,31HH - AVERAGE TRANSPORTATION COST,21X,F12.2,12H \$ PER SITE/	00000880
	+ 11X,41HMXHS - MAXIMUM ON-RADAR MAINTENANCE TIME,15X,F12.2,	00000890
	+ 20H MAN-HOURS PER SITE/	00000900
	+ 13X,46HNSP - NUMBER OF RADAR MAINTENANCE MEN PER SITE,	00000910
	+ 12X,15,8H MEN/	00000920
	+ 13X,52HSA - AVAILABLE WORK TIME FOR SITE RADAR MAINTENANCE,	00000930
	+ 2X,F12.2,33H HOURS PER YEAR PER SITE PER MAN/	00000940
	+ 11X,54HSMTBI - SYSTEM MEAN TIME BETWEEN MAINTENANCE INCIDENTS,	00000950
	+ 1X,F13.2,7H HOURS/	00000960
	+ 12X,55HSLB - YEARLY LABOR RATE FOR SITE RADAR MAINTENANCE MEN,	00000970
	+ F12.2,11H \$ PER MAN)	00000980
1600	FORMAT (1H0//7X,17HINVESTMENT SPARES//	00000990
	+ 15X,51HB - NUMBER OF STANDARD DEVIATIONS FOR CHF AND DEPOT/	00001000
	+ 19X,29H(ASSUMING EXPECTED BACKORDER),19X,F12.2/	00001010
	+ 12X,24HCONF - CONFIDENCE FACTOR,31X,F12.2/	00001020
	+ 12X,28HCRCT - CHF REPAIR CYCLE TIME,27X,F12.2,7H WEEKS/	00001030
	+ 12X,30HDRCT - DEPOT REPAIR CYCLE TIME,25X,F12.2,7H WEEKS/	00001040
	+ 13X,47HOST - ORDER AND SHIPPING TIME FROM DEPOT TO CHF,	00001050
	+ 7X,F12.2,7H WEEKS/	00001060
	+ 12X,55HSOSI - ORDER AND SHIPPING INTERVAL BETWEEN SITE AND CHF,	00001070
	+ F12.2,7H WEEKS/	00001080
	+ 12X,29HSRCT - SITE REPAIR CYCLE TIME,26X,F12.2,7H WEEKS)	00001090
1700	FORMAT (1H0/7X,20HINVENTORY MANAGEMENT//	00001100
	+ 13X,29HMC - INITIAL MANAGEMENT COST,25X,F12.2,12H \$ PER PART/	00001110
	+ 13X,31HRMC - RECURRING MANAGEMENT COST,23X,F12.2,	00001120
	+ 21H \$ PER PART PER YEAR/	00001130
	+ 14X,43HSA - SITE AND CHF INVENTORY MANAGEMENT COST,10X,F12.2,	00001140
	+ 30H \$ PER PART PER SITE PER YEAR)	00001150
1800	FORMAT (1H0/7X,20HSOFTWARE MAINTENANCE//	00001160
	+ 12X,32HCPWT - HARDWARE ENHANCEMENT COST,21X,F14.2,	00001170
	+ 12H \$ PER LBS./	00001180
	+ 11X,25HWPAC - SOFTWARE FACILITY,31X,F12.3,	00001190
	+ 15H \$ IN MILLIONS)	00001200
1900	FORMAT (1H0/7X,11HCONSUMABLES//	00001210
	+ 13X,34HPCS - COST OF POWER AT RADAR SITES,20X,F12.2,	00001220
	+ 35H \$ PER KILOWATT HOUR PER SITE /	00001230
	+ 12X,37HNCRS - MISCELLANEOUS CONSUMPTION RATE,18X,F12.2,	00001240
	+ 21H \$ PER YEAR PER SITE/	00001250
	+ 12X,38HPPRS - CONSUMPTION RATE OF PRIME POWER,17X,F12.2,	00001260

	+ 13H KW PER SITE/	00001270
	+ 12X,42HSPRS - CONSUMPTION RATE OF SECONDARY POWER,13X,F12.2,	00001280
	+ 13H KW PER SITE)	00001290
2000	FORMAT (1H0/7X,10HFACILITIES/)	00001300
2100	FORMAT (10X,4HFAC(,I1,21H) - COST OF FACILITY ,I2,29X,F12.3,	00001310
	+ 15H \$ IN MILLIONS)	00001320
2200	FORMAT (9X,4HFAC(,I2,21H) - COST OF FACILITY ,I2,29X,F12.3,	00001330
	+ 15H \$ IN MILLIONS)	00001340
2300	FORMAT (1H0/7X,8HTRAINING//	00001350
	+ 11X,39HCADRE - NUMBER OF NEM IN TRAINING CADRE,21X,I5,	00001360
	+ 5X,3HMEN/	00001370
	+ 13X,41HNCP - NUMBER OF CNP MAINTENANCE PERSONNEL,17X,I5,5X,	00001380
	+ 3HMEN/	00001390
	+ 13X,43HNDP - NUMBER OF DEPOT MAINTENANCE PERSONNEL,	00001400
	+ 15X,I5,5X,3HMEN/	00001410
	+ 14X,31HTC - COST OF MAINTENANCE COURSE,22X,F12.2,	00001420
	+ 20H \$ PER MAN PER WEEK/	00001430
	+ 11X,28HTCCAD - COST OF CADRE COURSE,28X,F12.2,	00001440
	+ 20H \$ PER MAN PER WEEK/	00001450
	+ 14X,40HTE - COST OF PECULIAR TRAINING EQUIPMENT,11X,	00001460
	+ F14.2,3H \$/	00001470
	+ 14X,38HTR - AVERAGE MAINTENANCE TURNOVER RATE,15X,F12.2/	00001480
	+ 14X,51HTW - AVERAGE TRAINING TIME FOR SITE, CNP, AND DEPOT,	00001490
	+ 2X,F12.2,7H WEEKS)	00001500
2400	FORMAT (11X,31HTWCAD - TRAINING TIME FOR CADRE,25X,F12.2,	00001510
	+ 7H WEEKS/	00001520
	+ 11X,20HYSCAD - CADRE SALARY,36X,F12.2,20H \$ PER MAN PER YEAR)	00001530
2500	FORMAT (1H1,38X,43HINPUT TABLE 1: MISCELLANEOUS SCALAR INPUTS,	00001540
	+ 12H (CONTINUED))	00001550
C		00001560
C		00001570
	WRITE(7,1000)	00001580
	WRITE(7,1100) K	00001590
	WRITE(7,1200) M,PIUP,XUC,YOH	00001600
	TEM1 = FSD / 1000000.	00001610
	TEM2 = PME / 1000000.	00001620
	TEM3 = PRODX / 1000000.	00001630
	TEM4 = REPURB / 1000000.	00001640
	TEM5 = SW / 1000000.	00001650
	TEM6 = VAL / 1000000.	00001660
	WRITE(7,1300) TEM1,TEM2,TEM3,TEM4,TEM5,TEM6	00001680
	WRITE(7,1400) ADCM,ADPH,CAA,CCMP,CDR,CDWH,CLR,CPHI	00001690
	WRITE(7,1500) CPMP,DAA,DLR,DS,H,MXHRS,NSP,SAA,SMTBI,YSLR	00001700
	WRITE(7,1700) INC,RMC,SA	00001710
	WRITE(7,2500)	00001720
	WRITE(7,1600) B,CONF,CRCT,DRCT,OST,SOSI,SRCT	00001730
	TEM1 = SWPAC / 1000000.	00001740
	WRITE(7,1800) CPWT,TEM1	00001750
	WRITE(7,1900) PCS,MCPS,PPRS,SPRS	00001760
	WRITE(7,2000)	00001770
C		00001780
	DO 50 J3=1,10	00001790
	IF (FAC(J3).LT.0.00001) GO TO 50	00001800
	TEM1 = FAC(J3) / 1000000.	00001810
	IF (J3.LE.9) WRITE(7,2100) J3,J3,TEM1	00001820
	IF (J3.GT.9) WRITE(7,2200) J3,J3,TEM1	00001830
50	CONTINUE	00001840
C		00001850
	WRITE(7,2300) CADRE,NCP,NDP,TC,TCCAD,TE,TR,TW	00001860
	WRITE(7,2400) TWCAD,YSCAD	00001870
C		00001880
	RETURN	00001890
	END	00001900

```

SUBROUTINE READ2                                00000010
C                                                00000020
C*****00000030
C* SUBROUTINE READ2 READS IN THE SUPPORT EQUIPMENT INPUTS FROM UNIT12.*00000040
C* ONE RECORD IS READ IN FOR EACH SE UP TO A MAXIMUM OF 150 SES IN *00000050
C* THE SYSTEM. NO INDEX NUMBER MAY EXCEED 150. *00000060
C* -A- INDICATES THE LARGEST SE INDEX THAT WAS INPUT. THE USER MUST *00000070
C* HAVE HIS SE INDICES IN ASCENDING ORDER, BUT INDEX NUMBERS MAY BE *00000080
C* SKIPPED (I.E., THERE MAY BE GAPS IN THE FILE). FURTHERMORE, THE *00000090
C* USER CAN HAVE THE PROGRAM SKIP A PARTICULAR SUPPORT EQUIPMENT TYPE *00000100
C* BY SETTING NSES,NSEC, AND NSED TO 0. FOR THAT SE TYPE. *00000110
C*****00000120
C                                                00000130
C      COMMON /RD2/ A,COC(150),COD(150),COS(150),INOSE(150),NSEC(150),
+      NSED(150),NSES(150),SEC(150),SENAME(150,24)
      REAL NSEC,NSED,NSES
      INTEGER A
C                                                00000140
C      COMMON /CNTL/ EXIT,ITER,MAXPMT,PRNT,REDO,REREAD
      INTEGER EXIT,PRNT,REDO,REREAD
C                                                00000150
C      COMMON /ERROR/ IERROR,IWARN
C                                                00000160
C      COMMON /INIT/ CONFLO,MAXFA,MAXLR,MAXSE
C                                                00000170
C      REAL NSECK,NSEDX,NSESX
      DATA STAR/1H*/
      DIMENSION SENAMX(24)
C                                                00000180
C
C      1  FORMAT (A1,I3,24A1,F4.0,F4.0,F4.0,F7.0,F3.2,F3.2,F3.2)
C      2  FORMAT (/49H UNIT 12 ERROR:  END OF FILE CARD NOT FOUND AFTER/
+      17X,40HMAXIMUM NUMBER OF SE TYPES WERE READ IN.)
C      3  FORMAT (A1)
C      4  FORMAT (/50H UNIT 12 ERROR:  SE NUMBERS NOT INCREASING FOR SE ,
+      6HNUMBER ,I3,1H.)
C      5  FORMAT (/48H UNIT 12 ERROR:  INDEX NUMBER WAS NOT POSITIVE. )
C      6  FORMAT (/49H UNIT 12 ERROR:  END OF FILE CARD NOT FOUND AFTER/
+      17X,40HMAXIMUM SE INDEX NUMBER WAS READ IN. )
C
C      A = 0
C
C*****00000470
C* THE SUPPORT EQUIPMENT INPUTS FOLLOW HERE. AN END-OF-FILE IS *00000480
C* DESIGNATED BY A STAR (*) IN COLUMN 1 OF THE LAST RECORD. IF MORE *00000490
C* THAN 150 SES ARE INPUT, AN ERROR MESSAGE IS PRINTED OUT. A *00000500
C* KEEPS TRACK OF THE LARGEST SE INDEX FROM UNIT 12. *00000510
C* USER MUST HAVE THE SE NUMBERS IN INCREASING ORDER, BUT GAPS MAY *00000520
C* EXIST. THE PROGRAM WILL KNOW THAT AN SE NUMBER WAS SKIPPED *00000530
C* BECAUSE NSEC+NSED+NSES FOR THAT SE INDEX WILL SUM TO ZERO. *00000540
C* NO INDEX NUMBER MAY BE LESS THAN ZERO OR GREATER THAN 150. *00000550
C*****00000560
C
C      DO 25 II=1,MAXSE
      READ (12,1) COL1,INOSE(II),(SENAME(J),J=1,24),
+      NSESX,NSECK,NSEDX,SECK,COSX,COCX,CODX
C
C.....SKIP OUT IF WE HAVE FOUND THE EOF MARKER.
      IF (COL1.EQ.STAR) GO TO 30

```

```

C.....SKIP THIS SE TYPE IF NSES,NSEC,NSED=0.                                00000640
      TOT = NSESI + NSECI + NSEDI                                           00000650
      IF (TOT.LT.0.0001) GO TO 25                                           00000660
      L=INOSI(II)                                                            00000670
C.....PRINT AN ERROR MESSAGE IF INDEX WAS NOT POSITIVE.                    00000680
      IF (L.GT.0) GO TO 21                                                  00000690
      WRITE(6,5)                                                            00000700
      IF (PRNT.NE.0) WRITE(7,5)                                             00000710
      IERROR = IERROR + 1                                                  00000720
      GO TO 25                                                              00000730
C.....ASSIGN VALUES FOR SE TYPE L.                                         00000740
21      DO 22 J=1,24                                                       00000750
      SENAME(L,J) = SENAMI(J)                                              00000760
22      CONTINUE                                                           00000770
      NSESI(L) = NSESI                                                    00000780
      NSECI(L) = NSECI                                                    00000790
      NSEDI(L) = NSEDI                                                    00000800
      SECI(L) = SECI                                                       00000810
      COSI(L) = COSI                                                       00000820
      COCI(L) = COCI                                                       00000830
      CODI(L) = CODI                                                       00000840
C.....PRINT AN ERROR MESSAGE IF INDEX DOESNT EXCEED PREVIOUS INDEX.        00000850
      IF (L.GT.A) GO TO 23                                                  00000860
      WRITE(6,4) L                                                         00000870
      IF (PRNT.NE.0) WRITE(7,4) L                                          00000880
      IERROR = IERROR + 1                                                  00000890
      GO TO 25                                                              00000900
23      A = L                                                                00000910
C.....PRINT AN ERROR MESSAGE IF INDEX EXCEEDS 150.                         00000920
      IF (A.LE.MAXSE) GO TO 24                                             00000930
      WRITE(6,6)                                                            00000940
      IF (PRNT.NE.0) WRITE(7,6)                                             00000950
      IERROR = IERROR + 1                                                  00000960
      GO TO 30                                                              00000970
24      IF (A.EQ.MAXSE) GO TO 26                                           00000980
25      CONTINUE                                                           00000990
C.....READ (12,3) COL1                                                     00010000
26      READ (12,3) COL1                                                  00010010
      IF (COL1.EQ.STAR) GO TO 30                                           00010020
      IF (II.GE.MAXSE) GO TO 27                                           00010030
C.....PRINT ERROR MESSAGE IF INDEX NUMBER 150 WAS NOT FOLLOWED BY EOF.    00010040
      WRITE (6,6)                                                          00010050
      IF (PRNT.NE.0) WRITE (7,6)                                           00010060
      GO TO 28                                                             00010070
C.....PRINT ERROR MESSAGE IF 151ST CARD NOT EOF.                          00010080
27      WRITE (6,2)                                                        00010090
      IF (PRNT.NE.C) WRITE (7,2)                                           00010100
28      IERROR=IERROR+1                                                    00010110
C.....RETURN                                                                00010120
30      RETURN                                                             00010130
      END                                                                00010140
      SUBROUTINE ITAB2                                                    00010150
C.....                                                                    00010160
C*****                                                                    00010170
C* SUBROUTINE ITAB2 PRINTS OUT TABLE 2 ON UNIT 7 (LINE PRINTER). IT *00010180
C* IS BASICALLY AN ECHO PRINT OF INPUT UNIT 12. *00010190
C*****                                                                    00010200
C.....                                                                    00010210
      COMMON /RD2/ A,COC(150),COD(150),COS(150),INOSI(150),NSEC(150),    00010220
      NSEDI(150),NSES(150),SECI(150),SENAI(150,24)                    00010230
      REAL NSEC,NSEDI,NSES                                               00010240
      INTEGER A                                                         00010250
      *****                                                            00010260

```

C		00001270
C		00001280
1	FORMAT (1H1,36X,29HINPUT TABLE 2: INPUTS BY SUP,	00001290
+	19HPORT EQUIPMENT TYPE///)	00001300
C		00001310
2	FORMAT (1H1,36X,41HINPUT TABLE 2: INPUTS BY SUPPORT EQUIPME,	00001320
+	19HNT TYPE (CONTINUED)///)	00001330
C		00001340
3	FORMAT (79X,8HFRACTION,5X,8HFRACTION,5X,8HFRACTION/	00001350
+	79X,9HUNIT COST,4X,9HUNIT COST,4X,9HUNIT COST/	00001360
+	1X,7HSUPPORT,29X,6HNUMBER,5X,6HNUMBER,4X,6HNUMBER,15X,	00001370
+	10HTO OPERATE,3X,10HTO OPERATE,3X,10HTO OPERATE/	00001380
+	1X,6HEQUIP.,30X,8HREQUIRED,3X,8HREQUIRED,	00001390
+	2X,8HREQUIRED,4X,4HUNIT,5X,	00001400
+	10H6 MAINTAIN,3X,10H6 MAINTAIN,3X,10H6 MAINTAIN/	00001410
+	1X,5HINDEX,6X,12HNOMENCLATURE,13X,9HEACH SITE,2X,6HAT CNF,4X,	00001420
+	8HAT DEPOT,4X,4HCOST,5X,7HAT SITE,6X,6HAT CNF,7X,8HAT DEPOT/	00001430
+	37X,6H(NSES),5X,6H(NSEC),4X,6H(NSED),6X,5H(SEC),4X,5H(COS),	00001440
+	8X,5H(COC),8X,5H(COD)/)	00001450
C		00001460
4	FORMAT (2X,I3,4X,24A1,4X,F7.2,4X,F6.2,4X,F6.2,4X,F7.0,4X,F5.2,	00001470
+	8X,F5.2,8X,F5.2)	00001480
C		00001490
C*****		00001500
C* PRINT OUT TABLE 2 HEADINGS.		*00001510
C*****		00001520
C		00001530
	LINCNT=0	00001540
	WRITE (7,1)	00001550
	WRITE (7,3)	00001560
C		00001570
C*****		00001580
C* THIS DO-LOOP PRINTS OUT THE UNIT 12 DATA. 50 LINES OF DATA ARE		*00001590
C* WRITTEN BEFORE THE PRINTER JUMPS TO A NEW PAGE. TABLE HEADINGS		*00001600
C* ARE PRINTED ON EACH NEW PAGE.		*00001610
C*****		00001620
C		00001630
	IF (A.EQ.0) GO TO 26	00001640
	DO 25 L=1,A	00001650
	TOT = NSES(L) + NSEC(L) + NSED(L)	00001660
	IF (TOT.LT.0.0001) GO TO 25	00001670
	WRITE (7,4) L, (SENAME(L,J),J=1,24), NSES(L), NSEC(L), NSED(L),	00001680
+	SEC(L), COS(L), COC(L), COD(L)	00001690
C		00001700
	LINCNT=LINCNT+1	00001710
	IF (LINCNT.LT.50) GO TO 25	00001720
	WRITE (7,2)	00001730
	WRITE (7,3)	00001740
	LINCNT=0	00001750
25	CONTINUE	00001760
26	CONTINUE	00001770
C		00001780
	RETURN	00001790
	END	00001800


```

SUBROUTINE READ3
C
C*****
C* SUBROUTINE READ3 READS IN THE FUNCTIONAL AREA INPUTS FROM UNIT 13.
C* ONE RECORD IS READ IN FOR EACH PA UP TO A MAXIMUM OF 10 PA'S IN
C* THE SYSTEM.
C* THE INDEX NUMBERS MUST BE CONSECUTIVE STARTING WITH 1.
C*****
COMMON /RD3/ ENYR(10),PA,PANAME(10,24),HWCF(10),INOPA(10),MMH(10),
+ SIZE(10),SMI(10),SWCF(10),SWFIX(10),SWVAR(10),
+ WEIGHT(10)
INTEGER PA
REAL MMH
C
COMMON /CNTL/ EXIT,ITER,MAXPMT,PRNT,REDO,REREAD
INTEGER EXIT,PRNT,REDO,REREAD
C
COMMON /ERROR/ IERROR,IWARN
C
COMMON /INIT/ CONFLO,MAXPA,MAXLR,MAXSE
C
DATA STAR/1H*/
C
1 FORMAT (A1,I3,24A1,F5.0,2F5.2,F7.0,F3.2,F7.0,F3.2,F3.1,F6.3)
C
2 FORMAT (/49H UNIT 13 ERROR: END OF FILE CARD NOT FOUND AFTER/
+ 17X,41HMAXIMUM NUMBER OF PA TYPES WERE READ IN.)
C
3 FORMAT (A1)
C
4 FORMAT (/51H UNIT 13 ERROR: PA NUMBERS NOT CONSECUTIVE STARTI,
+ 25HNG WITH 1 FOR PA NUMBER ,I3,1H.)
C
PA = 0
C*****
C* THE FUNCTIONAL AREA FILE INPUTS IN THIS LOOP. AN END-OF-FILE IS
C* DESIGNATED BY A STAR (*) IN COLUMN 1 OF THE LAST RECORD. IF MORE
C* THAN 10 PAS ARE INPUT, AN ERROR MESSAGE IS PRINTED OUT. PA
C* KEEPS TRACK OF THE LARGEST PA INDEX FROM UNIT 12.
C* USER MUST HAVE THE PA NUMBERS CONSECUTIVE STARTING WITH 1.
C*****
DO 25 J=1,MAXPA
READ (13,1) COL1,INOPA(J),(PANAME(J,L),L=1,24),SMI(J),MMH(J),
+ ENYR(J),WEIGHT(J),HWCF(J),SIZE(J),SWCF(J),SWFIX(J),SWVAR(J)
C
IF (COL1.EQ.STAR) GO TO 30
IF (INOPA(J).EQ.J) GO TO 23
WRITE (6,4) INOPA(J)
IF (PRNT.NE.0) WRITE(7,4) INOPA(J)
IERROR = IERROR + 1
23 CONTINUE
PA = INOPA(J)
SWFIX(J) = SWFIX(J) * 1000.
25 CONTINUE
C
READ (13,3) COL1
IF (COL1.EQ.STAR) GO TO 30
WRITE (6,2)
IF (PRNT.NE.0) WRITE (7,2)

```

```

C      IEPROR=IERROR+1                                00000640
C      30      RETURN                                    00000650
C      END                                            00000660
C      SUBROUTINE ITAB3                                00000670
C                                                    00000680
C                                                    00000690
C*****00000700
C* SUBROUTINE ITAB3 PRINTS INPUT TABLE 3, INPUTS BY FUNCTIONAL AREA. *00000710
C*****00000720
C                                                    00000730
C      COMMON /RD3/ ENYR(10),FA,FANAME(10,24),HWCP(10),INOPA(10),MMH(10),00000740
C      +      SIZE(10),SMI(10),SWCP(10),SWFIX(10),SWVAR(10),00000750
C      +      WEIGHT(10)                                00000760
C      INTEGER FA                                        00000770
C      REAL MMH                                          00000780
C                                                    00000790
C                                                    00000800
C      1  FORMAT (1H1,30X,41HINPUT TABLE 3: INPUTS BY FUNCTIONAL AREA//00000810
C      +      33X,49HAVG. HRS. MAN-HOURS EXPECTED WEIGHT EXPECTED ,00000820
C      +      17HNUMBER EXPECTED,10X,9HMAN-POWER/00000830
C      +      6H PUNC-,27X,38HMAINT. SCHEDULED NUMBER IN LBS. ,00000840
C      +      46HFRACTION SOFTWARE FRACTION FIXED COST PER/00000850
C      +      7H TIONAL,26X,37HINTERVAL MAINT. ENHANCE- HARDWARE,00000860
C      +      45H HARDWARE OBJECT SOFTWARE MAN- OBJECT/00000870
C      +      5H AREA,26X,38HBY SITE TASK SITE MENTS AND ,00000880
C      +      46HCHANGED INSTRUC- CHANGED POWER INSTRUC-/00000890
C      +      22H INDEX NOMENCLATURE,11X,00000900
C      +      54HPERSONNEL PERSONNEL PER YEAR FIRMWARE PER ENHMT. TIONS,00000910
C      +      26H PER ENHMT. COST TION/00000920
C      +      33X,5H(SMI),5X,5H(MMH),5X,6H(ENYR),3X,15H(WEIGHT) (HWCP),00000930
C      +      5X,6H(SIZE),3X,6H(SWCP),4X,15H(SWFIX) (SWVAR) /00000940
C      2  FORMAT (1X,I3,4X,24A1,F7.2,F10.2,F9.0,F10.0,F9.2,F11.0,00000950
C      +      F9.2,F10.1,F10.3)00000960
C                                                    00000970
C                                                    00000980
C      WRITE(7,1)00000990
C      DO 5 J=1,FA00010000
C      +      WRITE(7,2) INOPA(J), (FANAME(J,L),L=1,24),SMI(J),MMH(J),ENYR(J),00010100
C      +      WEIGHT(J),HWCP(J),SIZE(J),SWCP(J),SWFIX(J),SWVAR(J)00010120
C      5  CONTINUE00010130
C                                                    00010140
C      RETURN00010150
C      END00010160

```

```

SUBROUTINE READ4
C*****00000010
C00000020
C*****00000030
C* SUBROUTINE READ4 READS IN THE FIRST LRU FILE OFF OF INPUT UNIT 14. 00000040
C* ONE RECORD IS READ IN FOR EACH LRU UP TO A MAXIMUM OF 200 LRUS IN *00000050
C* THE SYSTEM. 00000060
C*****00000070
C00000080
COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200),
+ MTBI(200),N,PA(200),QPA(200),QR(200),RM(200),
+ UC(200) 00000090
INTEGER QPA,QR 00000100
REAL MTBI 00000110
C00000120
C00000130
COMMON /ERROR/ IERROR,IWARN 00000140
C00000150
COMMON /INIT/ CONFLO,MAXPA,MAXLR,MAXSE 00000160
C00000170
COMMON /CNTL/ EXIT,ITER,MAXPNT,PRNT,REDO,BEREAD 00000180
C00000190
INTEGER EXIT,PRNT,REDO,BEREAD 00000200
C00000210
REAL MTBIX 00000220
DATA STAR/1H*/ 00000230
DIMENSION LRNAMEX(24),LRPARX(12) 00000240
INTEGER QPAX,QRX 00000250
C00000260
1 FORMAT (A1,I3,2A1,12A1,I3,I3,F7.0,F3.3,F9.0,F5.2,I1) 00000270
C00000280
2 FORMAT (/49H UNIT 14 ERROR: END OF FILE CARD NOT FOUND AFTER/ 00000290
+ 17X,41HMAXIMUM NUMBER OF LRU TYPES WERE READ IN.) 00000300
C00000310
3 FORMAT (A1) 00000320
C00000330
4 FORMAT (/52H UNIT 14 ERROR: LRU NUMBERS NOT INCREASING FOR LRU , 00000340
+ 7HNUMBER ,I3,1H.) 00000350
C00000360
5 FORMAT (/48H UNIT 14 ERROR: INDEX NUMBER WAS NOT POSITIVE. ) 00000370
C00000380
6 FORMAT (/49H UNIT 14 ERROR: END OF FILE CARD NOT FOUND AFTER/ 00000390
+ 17X,41HMAXIMUM LRU INDEX NUMBER WAS READ IN. ) 00000400
C00000410
N = 0 00000420
C00000430
C*****00000440
C* THE FIRST LRU FILE IS INPUT IN THIS DO-LOOP. AN END-OF-FILE IS *00000450
C* DESIGNATED BY A STAR (*) IN COLUMN 1 OF THE LAST RECORD. IF MORE *00000460
C* THAN 200 LRUS ARE INPUT, AN ERROR MESSAGE IS PRINTED OUT. N *00000470
C* KEEPS TRACK OF THE LARGEST LRU INDEX FROM UNIT 14. *00000480
C* THE LARGEST LRU INDEX MUST NOT EXCEED 200. *00000490
C* USER MUST HAVE THE LRU NUMBERS IN INCREASING ORDER, BUT GAPS MAY *00000500
C* EXIST. THE PROGRAM WILL KNOW THAT AN LRU NUMBER WAS SKIPPED *00000510
C* BECAUSE QPA FOR THAT LRU NUMBER WILL HAVE A VALUE OF ZERO. *00000520
C*****00000530
C00000540
DO 25 II=1,MAXLR 00000550
READ (14,1) COL1,INO(II),(LRNAMEX(J),J=1,24),(LRPARX(J),
+ J=1,12),QPAX,QRX,UCK,RMX,MTBIX,PAX,MCIX 00000560
C00000570
C00000580
C.....SKIP OUT OF ROUTINE IF WE HAVE COME TO THE EOF CARD. 00000590
IF (COL1.EQ.STAR) GO TO 30 00000600
I=INO(II) 00000610
C.....PRINT ERROR MESSAGE IF INDEX NOT POSITIVE. 00000620
IF (I.GT.0) GO TO 20 00000630

```



```

WRITE(6,5) 00000640
IF (PRNT.NE.0) WRITE(7,5) 00000650
IERROR = IERROR + 1 00000660
GO TO 25 00000670
C.....PRINT ERROR MESSAGE IF INDEX GREATER THAN 200. 00000680
20 IF (I.LE.MAXLR) GO TO 21 00000690
WRITE(6,6) 00000700
IF (PRNT.NE.0) WRITE(7,6) 00000710
IERROR = IERROR + 1 00000720
GO TO 30 00000730
C.....ASSIGN VALUES TO LRU VARIABLES. 00000740
21 DO 22 J=1,24 00000750
LRNAME(I,J) = LRNAME(J) 00000760
IF (J.LT.13) LRPART(I,J) = LRPART(J) 00000770
22 CONTINUE 00000780
QPA(I) = QPA 00000790
QR(I) = QR 00000800
UC(I) = UC 00000810
RM(I) = RM 00000820
MTBI(I) = MTBI 00000830
PA(I) = PA 00000840
NCI(I) = NCI 00000850
C.....PRINT ERROR MESSAGE IF INDEX NOT GREATER THAN PREVIOUS INDEX. 00000860
IF (I.GT.N) GO TO 23 00000870
WRITE(6,4) I 00000880
IF (PRNT.NE.0) WRITE(7,4) I 00000890
IERROR = IERROR + 1 00000900
GO TO 25 00000910
23 N = I 00000920
IF (N.EQ.MAXLR) GO TO 26 00000930
25 CONTINUE 00000940
C 00000950
26 READ (14,3) COL1 00000960
IF (COL1.EQ.STAR) GO TO 30 00000970
IF (II.GE.MAXLR) GO TO 27 00000980
C.....PRINT ERROR MESSAGE IF EOP NOT FOUND AFTER INDEX 200. 00000990
WRITE (6,6) 00001000
IF (PRNT.NE.0) WRITE (7,6) 00001010
GO TO 28 00001020
C.....PRINT ERROR MESSAGE IF MORE THAN 200 LRUS IN FILE. 00001030
27 WRITE (6,2) 00001040
IF (PRNT.NE.0) WRITE (7,2) 00001050
28 IERROR=IERROR+1 00001060
C 00001070
30 RETURN 00001080
END 00001090
SUBROUTINE ITAB4 00001100
C 00001110
C***** 00001120
C* SUBROUTINE ITAB4 PRINTS OUT TABLE 4 ON UNIT 7 (LINE PRINTER). IT *00001130
C* IS BASICALLY AN ECHO PRINT OF INPUT UNIT 14. *00001140
C***** 00001150
COMMON /RD4/ INO(200), LRNAME(200,24), LRPART(200,12), NCI(200), 00001160
+ MTBI(200), N, PA(200), QPA(200), QR(200), RM(200), 00001170
+ UC(200) 00001180
INTEGER QPA,QR 00001190
REAL MTBI 00001200
C 00001210
1 FORMAT (1H1,46X,29HINPUT TABLE 4: INPUTS BY LRU///) 00001220
C 00001230
2 FORMAT (1H1,40X,41HINPUT TABLE 4: INPUTS BY LRU (CONTINUED)///) 00001240
C 00001250
00001260

```



```

3  FORMAT (47X,6HNUMBER,3X,9HNUMBER OF,11X,8HFRACTION,3X,9HMEAN TIME,00001270
+ 2X,9HNUMBER OF/47X,18HOF LRUS REDUNDANT,11X,9HUNIT COST, 00001280
+ 9H BETWEEN,4X,10HNEW ASSEN-,3X,7HMISSION/5H LRU,42X,7HIN EACH00001290
+ 9H LRUS IN,6X,4HUNIT,3X,8HCONSUMED,3X,6HHAINT.,5X,8HBLIES OR,00001300
+ 5X,8HCRITICAL/6H INDEX,6X,12HNONENCLATURE,9X,11HPART NUMBER, 00001310
+ 8H RADAR,4X,10HEACH RADAR,3X,4HCOST,3X,18HIN REPAIR INCIDEN,00001320
+ 2HTS,2X,22HPIECE PARTS INDICATOR/6H (INO),41X,5H(QPA),4X, 00001330
+ 4H(QR),9X,4H(UC),3X,4H(RN),7X,6H(MTBI),5X,4H(PA),9X,5H(NCI)/) 00001340
C 00001350
4  FORMAT (2X,I3,3X,24A1,1X,12A1,3X,I3,7X,I3,6X,F8.0,3X,F5.3,2X, 00001360
+ F10.0,5X,F6.2,8X,I2) 00001370
C 00001380
C*****00001390
C* PRINT OUT TABLE 4 HEADINGS. *00001400
C*****00001410
C 00001420
C LINCNT=0 00001430
C WRITE (7,1) 00001440
C WRITE (7,3) 00001450
C 00001460
C*****00001470
C* THIS DO-LOOP PRINTS OUT THE UNIT 14 DATA. 50 LINES OF DATA ARE *00001480
C* WRITTEN BEFORE THE PRINTER JUMPS TO A NEW PAGE. TABLE HEADINGS *00001490
C* ARE PRINTED ON EACH NEW PAGE. *00001500
C*****00001510
C 00001520
C DO 25 I=1,N 00001530
C IF (QPA(I).EQ.0) GO TO 25 00001540
C WRITE (7,4) I,(LRNAME(I,J),J=1,24),(LRPART(I,J),J=1,12), 00001550
+ QPA(I),QR(I),UC(I),RN(I),MTBI(I),PA(I),NCI(I) 00001560
C 00001570
C LINCNT=LINCNT+1 00001580
C IF (LINCNT.LT.50) GO TO 25 00001590
C WRITE (7,2) 00001600
C WRITE (7,3) 00001610
C LINCNT=0 00001620
25 CONTINUE 00001630
C 00001640
C RETURN 00001650
C END 00001660

```

```

SUBROUTINE READ5
C
C*****00000010
C*****00000020
C*****00000030
C* SUBROUTINE READ5 READS IN THE SECOND LRU FILE OFF OF INPUT UNIT *00000040
C* 15. ONE RECORD IS READ IN FOR EACH LRU UP TO A MAXIMUM OF 200 *00000050
C* LRUS IN THE SYSTEM. *00000060
C*****00000070
C
COMMON /CNTL/ EXIT,ITER,MAXPNT,PRNT,REDO,REREAD
INTEGER EXIT,PRNT,REDO,REREAD
C
COMMON /ERROR/ IERROR,IWARN
C
COMMON /INIT/ CONFLO,MAXPA,MAXLR,MAXSE
C
COMMON /RD4/ INO(200),LFNAME(200,24),LRPART(200,12),MCI(200),
+ MTBI(200),N,PA(200),QPA(200),QR(200),RH(200),
+ UC(200)
INTEGER QPA,QR
REAL MTBI
C
COMMON /RD5/ BCMH(200),CBCMH(200),CMH(200),CRTS1(200),CRTS2(200),
+ DBCM(200),DMH(200),DRTS1(200),DRTS2(200),
+ DRTS3(200),PPR(200),IMH(200),INO15(200),N15,
+ PAMH(200),RIP(200),RL(200),RMH(200),SMH(200),SRTS1(200),
+ WOR1(200),WOR2(200),WOR3(200)
REAL IMH
INTEGER RL
C
DATA STAR/1H*/
REAL IMHX
INTEGER RLX
C
1 FORMAT (A1,I3,F3.3,19(F3.2),5X,I3)
C
2 FORMAT (/48H UNIT 15 ERROR: LRU INDEX NUMBERS HAVE NOT BEEN/
+ 17X,44HINPUT IN THE SAME ORDER FOR UNITS 14 AND 15./
+ 17X,29HUNIT 15 LRU INDEX NUMBER WAS ,I3,1H.)
C
3 FORMAT (/49H UNIT 15 ERROR: END OF FILE CARD NOT FOUND AFTER/
+ 17X,37HMAXIMUM NUMBER OF LRUS WERE READ IN. )
C
4 FORMAT (A1)
C
5 FORMAT (/52H UNIT 15 ERROR: LARGEST LRU NUMBER INPUT ON UNIT 14
+ ,13H IS DIFFERENT/17X,36HTHAN THE LARGEST LRU NUMBER INPUT ON,
+ 9H UNIT 15.)
C
6 FORMAT (/48H UNIT 15 ERROR: INDEX NUMBER WAS NOT POSITIVE. )
C
7 FORMAT (/49H UNIT 15 ERROR: END OF FILE CARD NOT FOUND AFTER/
+ 17X,37HMAXIMUM LRU INDEX NUMBER WAS READ IN.)
C
N15=0
C
C*****00000050
C* THE SECOND LRU FILE IS INPUT IN THIS DO-LOOP. AN END-OF-FILE IS *000000570
C* DESIGNATED BY A STAR (*) IN COLUMN 1 OF THE LAST RECORD. *000000580
C*****000000590
C
DO 25 II=1,MAXLR
READ (15,1) COL1,INO15(II),PPRX,PAMHX,RIPX,IMHX,RMHX,BCMHX,
+ SMHX,CBCMHX,CMHX,DBCMHX,DMHX,SRTS1X,CRTS1X,
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630

```

+	WOR1X, DRTS1X, CRTS2X, WOR2X, DRTS2X, WOR3X, DRTS3X,	00000640
+	RLX	00000650
C		00000660
	IF (COL1.EQ.STAR) GO TO 30	00000670
	I=INO15(II)	00000680
C.....	PRINT ERROR MESSAGE IF INDEX NOT POSITIVE.	00000690
	IF (I.GT.0) GO TO 20	00000700
	WRITE(6,6)	00000710
	IF (PRNT.NE.0) WRITE(7,6)	00000720
	IERROR = IERROR + 1	00000730
	GO TO 25	00000740
C.....	PRINT ERROR MESSAGE IF INDEX EXCEEDS 200.	00000750
20	IF (I.LE.MAXLR) GO TO 21	00000760
	WRITE(6,7)	00000770
	IF (PRNT.NE.0) WRITE(7,7)	00000780
	IERROR = IERROR + 1	00000790
	GO TO 30	00000800
C		00000810
C*****		00000820
C*	IF THE LRUS READ IN HERE ARE NOT INPUT IN THE EXACT SAME ORDER AS	*00000830
C*	THE LRUS ON UNIT 14, AN ERROR MESSAGE IS PRINTED FOR EACH LRU OUT	*00000840
C*	OF ORDER.	*00000850
C*****		00000860
C		00000870
21	IF (INO15(II).EQ.INO(II)) GO TO 22	00000880
	WRITE(6,2) INO15(II)	00000890
	IF (PRNT.NE.0) WRITE(7,2) INO15(II)	00000900
	IERROR=IERROR+1	00000910
C		00000920
22	N15=INO15(II)	00000930
	FPR(I) = FPRX	00000940
	PAMH(I) = PAMHX	00000950
	RIP(I) = RIPX	00000960
	IMH(I) = IMHX	00000970
	RMH(I) = RMHX	00000980
	BCMHI(I) = BCMHX	00000990
	SMH(I) = SMHX	00010000
	CBCMH(I) = CBCMHX	00010010
	CMH(I) = CMHX	00010020
	DBCMI(I) = DBCMHX	00010030
	DMH(I) = DMHX	00010040
	SRTS1(I) = SRTS1X	00010050
	CRTS1(I) = CRTS1X	00010060
	WOR1(I) = WOR1X	00010070
	DRTS1(I) = DRTS1X	00010080
	CPTS2(I) = CRTS2X	00010090
	WOR2(I) = WOR2X	00010100
	DRTS2(I) = DRTS2X	00010110
	WOR3(I) = WOR3X	00010120
	DRTS3(I) = DRTS3X	00010130
	RL(I) = RLX	00010140
	IF (N15.EQ.MAXLR) GO TO 26	00010150
C		00010160
25	CONTINUE	00010170
C		00010180
26	READ(15,4) COL1	00010190
	IF (COL1.EQ.STAR) GO TO 30	00010200
	IF (II.GE.MAXLR) GO TO 27	00010210
C.....	PRINT ERROR IF EOF NOT FOUND AFTER INDEX 200.	00010220
	WRITE(6,7)	00010230
	IF (PRNT.NE.0) WRITE(7,7)	00010240
	GO TO 28	00010250
C.....	PRINT ERROR MESSAGE IF MORE THAN 200 LRUS IN FILE.	00010260


```

27      WRITE (6,3)                                00001270
      IF (PRNT.NE.0) WRITE (7,3)                    00001280
28      IERROR=IERROR+1                             00001290
C                                              00001300
C*****00001310
C* N15 KEEPS TRACK OF THE LARGEST LRU NUMBER FROM UNIT 15. IF THIS IS *00001320
C* NOT THE SAME AS N (LARGEST LRU NUMBER FROM UNIT 14), AN ERROR *00001330
C* MESSAGE IS PRINTED OUT. *00001340
C*****00001350
C                                              00001360
30      IF (N.EQ.N15) GO TO 40                      00001370
      WRITE (6,5)                                    00001380
      IF (PRNT.NE.0) WRITE (7,5)                    00001390
      IERROR=IERROR+1                               00001400
C                                              00001410
40      RETURN                                       00001420
      END                                           00001430
      SUBROUTINE ITAB5                               00001440
C                                              00001450
C*****00001460
C* SUBROUTINE ITAB5 PRINTS OUT TABLE 5 PART 1 AND TABLE 5 PART 2 ON *00001470
C* UNIT 7 (LINE PRINTER). IT IS BASICALLY ON ECHO PRINT OF INPUT *00001480
C* UNIT 15. *00001490
C*****00001500
C                                              00001510
      COMMON /INIT/ CONFLO,MAXPA,MAXLR,MAXSE        00001520
C                                              00001530
C      COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),NCI(200),
+      MTBI(200),N,PA(200),QPA(200),QR(200),RM(200),
+      UC(200)                                       00001560
      INTEGER QPA,QR                                00001570
      REAL MTBI                                     00001580
C                                              00001590
C      COMMON /RD5/ BCBMH(200),CBCMH(200),CHH(200),CRTS1(200),CRTS2(200),
+      DBCMH(200),DMH(200),DRTS1(200),DRTS2(200),
+      DRTS3(200),PPR(200),INH(200),INO15(200),N15,
+      PAMH(200),RIP(200),RL(200),RMH(200),SMH(200),SRTS1(200),
+      WOR1(200),WOR2(200),WOR3(200)                00001640
      REAL INH                                       00001650
      INTEGER RL                                     00001660
C                                              00001670
C      1  FORMAT (1H1,37X,45HINPUT TABLE 5 PART 1: MAINTENANCE INPUTS BY , 00001680
+      3HLRU///)                                     00001690
C                                              00001700
C      2  FORMAT (1H1,31X,45HINPUT TABLE 5 PART 1: MAINTENANCE INPUTS BY , 00001710
+      15HLRU (CONTINUED) ///)                      00001720
C                                              00001730
C      3  FORMAT (15X,7HAVERAGE,12X,25HAVERAGE AVG. MAN- AVERAGE,
+      4X,7HAVERAGE,
+      4X,7HAVERAGE,4X,7HAVERAGE,4X,7HAVERAGE/    00001760
+      15X,18HMAN-HOURS FRACTION,28H MAN-HRS HRS FAULT MAN-HOURS,
+      2X,9HMAN-HOURS,11H MAN-HOURS,11H MAN-HOURS,11H MAN-HOURS,
+      11H MAN-HOURS/8X,5HFALSE,2X,9HFOR PREP-,9H FAILURES,1X,
+      6HFOR IN,2X,8HISOLATE,,2X,8HFOR SITE,3X,8HFOR SITE,3X,
+      7HFOR CMP,
+      4X,7HFOR CMP,4X,9HFOR DEPOT,2X,9HFOR DEPOT/5H LRU,3X,
+      4HPULL,3X,9HARATION &,9H REPAIRED,6H PLACE,3X,8HREPLACE,,2X,
+      10HSHOP BENCH,11H CORRECTIVE,11H SHOP BENCH,11H CORRECTIVE,
+      11H SHOP BENCH,11H CORRECTIVE/6H INDEX,6H RATE,
+      3X,6HACCESS,4X,8HIN PLACE,7H REPAIR,17H & VERIFY CHECK,
+      6X,6HMAINT.,5X,5HCHECK,6X,6HMAINT.,5X,5HCHECK,6X,6HMAINT./
+      6H (INO),2X,5H (PPR),2X,6H (PAMH),4X,5H (RIP),4X,5H (INH),3X,
+      5H (RMH),5X,6H (BCMH),5X,5H (SMH),6X,7H (CBCMH),4X,5H (CHH),6X,
+      00001890

```



```

      + 7H(DBCMH),4X,5H(DMH)/)                                00001900
C                                                                    00001910
4  FORMAT (2X,I3,3X,P5.3,3X,P5.3,5X,P5.3,4X,P5.3,3X,P5.3,5X,P5.3,6X, 00001920
      + P5.3,6X,P5.3,6X,P5.3,6X,P5.3,6X,P5.3)                00001930
C                                                                    00001940
5  FORMAT (1H1,36X,46HINPUT TABLE 5 PART 2: REPAIR LEVEL INPUTS BY ,00001950
      + 3HLRU//)                                                00001960
C                                                                    00001970
6  FORMAT (1H1,36X,46HINPUT TABLE 5 PART 2: REPAIR LEVEL INPUTS BY ,00001980
      + 15HLRU (CONTINUED)///)                                  00001990
C                                                                    00002000
7  FORMAT (9X,39H-----IF SITE REPAIRABLE-----,4X,        00002010
      + 29H-----IF CMF REPAIRABLE-----,4X,19HIF DEPOT REPAIRABLE/ 00002020
      + 25X,6H(RL=1),32X,6H(RL=2),22X,6H(RL=3)/9X,8HFRACTION,        00002030
      + 10H FRACTION,10H FRACTION,11H FRACTION,4X,8HFRACTION,        00002040
      + 10H FRACTION,11H FRACTION,4X,8HFRACTION,11H FRACTION/        00002050
      + 9X,8HFAILURES,10H FAILURES,10H FAILURES,11H FAILURES,4X,    00002060
      + 8HFAILURES,10H FAILURES,11H FAILURES,4X,8HFAILURES,        00002070
      + 11H FAILURES/5H LRU,4X,8HREPAIRED,10H REPAIRED,            00002080
      + 11H CONDEMNED,10H REPAIRED,4X,8HREPAIRED,11H CONDEMNED,    00002090
      + 10H REPAIRED,4X,9HCONDEMNED,10H REPAIRED,4X,6HREPAIR/      00002100
      + 6H INDEX,3X,7HAT SITE,3X,6HAT CMF,4X,6HAT CMF,5X,          00002110
      + 8HAT DEPOT,4X,6HAT CMF,4X,6HAT CMF,5X,8HAT DEPOT,4X,        00002120
      + 8HAT DEPOT,3X,8HAT DEPOT,4X,5HLEVEL/6H (INO),3X,7H(SRTS1), 00002130
      + 3X,7H(CRTS1),3X,6H(WOR1),5X,7H(DRTS1),5X,7H(CRTS2),3X,     00002140
      + 6H(WOR2),5X,7H(DRTS2),5X,6H(WOR3),5X,7H(DRTS3),5X,4H(RL)/ 00002150
C                                                                    00002160
8  FORMAT (2X,I3,5X,P5.3,5X,P5.3,5X,P5.3,6X,P5.3,7X,P5.3,5X,P5.3, 00002170
      + 6X,P5.3,7X,P5.3,7X,P5.3,5X,I3)                        00002180
C                                                                    00002190
C*****00002200
C* PRINT OUT TABLE 5 PART 1 HEADINGS.                          *00002210
C*****00002220
C                                                                    00002230
C  LINCNT=0                                                       00002240
      WRITE (7,1)                                                  00002250
      WRITE (7,3)                                                  00002260
C                                                                    00002270
C*****00002280
C* THIS DO-LOOP PRINTS OUT HALF OF THE UNIT 15 INPUT DATA. 50 LINES *00002290
C* OF DATA ARE WRITTEN BEFORE THE PRINTER JUMPS TO A NEW PAGE. TABLE *00002300
C* HEADINGS ARE PRINTED ON EACH NEW PAGE.                        *00002310
C*****00002320
C                                                                    00002330
C  DO 25 I=1,N15                                                  00002340
      IF (QPA(I).EQ.0) GO TO 25                                     00002350
      WRITE (7,4) I,FPR(I),PAHH(I),RIP(I),IMH(I),RMH(I),BCMH(I),    00002360
      + SMH(I),CBCMH(I),CMH(I),DBCMH(I),DMH(I)                   00002370
C                                                                    00002380
C  LINCNT=LINCNT+1                                                00002390
      IF (LINCNT.LT.50) GO TO 25                                  00002400
      WRITE (7,2)                                                  00002410
      WRITE (7,3)                                                  00002420
      LINCNT=0                                                      00002430
25  CONTINUE                                                       00002440
C                                                                    00002450
C*****00002460
C* PRINT OUT TABLE 5 PART 2 HEADINGS.                          *00002470
C*****00002480
C                                                                    00002490
C* LINCNT=0                                                       00002500
      WRITE (7,5)                                                  00002510
      WRITE (7,7)                                                  00002520

```

C		00002530
C*****		00002540
C*	THIS DO-LOOP PRINTS OUT THE SECOND HALF OF THE UNIT 15 INPUT DATA.	*00002550
C*	ONCE AGAIN, 50 LINES OF DATA ARE PRINTED PER PAGE.	*00002560
C*****		00002570
C*		00002580
	DO 50 I=1,N15	00002590
	IF (QPA(I).EQ.0) GO TO 50	00002600
	WRITE (7,8) I,SRTS1(I),CRTS1(I),WOR1(I),DRTS1(I),	00002610
+	CRTS2(I),WOR2(I),DRTS2(I),WOR3(I),DRTS3(I),RL(I)	00002620
C		00002630
	LINCNT=LINCNT+1	00002640
	IF (LINCNT.LT.50) GO TO 50	00002650
	WRITE (7,6)	00002660
	WRITE (7,7)	00002670
	LINCNT=0	00002680
50	CONTINUE	00002690
C		00002700
	RETURN	00002710
	END	00002720

```

SUBROUTINE ERRCHK
C
C*****
C* SUBROUTINE ERRCHK
C* THIS SUBROUTINE PERFORMS SOME ADDITIONAL ERROR CHECKS ON THE
C* INPUT DATA. THE FOLLOWING ARE ERRORS:
C* 1- RL < 0 OR RL > 3 FOR AN LRU.
C* 2- SRTS1 + CRTS1 + WOR1 + DRTS1 = 1 FOR LRU WITH RL OF 1.
C* 3- CRTS2 + WOR2 + DRTS2 = 1 FOR LRU WITH RL OF 2.
C* 4- WOR3 + DRTS3 = 1 FOR LRU WITH RL OF 3.
C* 5- MTBI = 0 FOR AN LRU.
C* 6- SMTBI = 0 OR
C* (1./SMTBI) < SUM(QPA(I)/MTBI(I),I=1,N) CAUSES A WARNING
C* MESSAGE AND CAUSES SMTBI TO BE ASSIGNED A DEFAULT VALUE OF
C* 1./ (SUM(QPA(I)/MTBI(I),I=1,N)).
C* 7- CPNI = 0 CAUSES A WARNING MESSAGE AND AN ASSUMED INFINITE
C* VALUE.
C* 8- CONF < CONFLO CAUSES A WARNING MESSAGE AND A DEFAULT VALUE
C* FOR CONF OF CONFLO (CONFLO IF SET TO 0.84134 IN INITIAL).
C* 9- NSP > 3 CAUSES A WARNING MESSAGE AND A DEFAULT VALUE OF 3.
C*****
COMMON /CNTL/ EXIT,ITER,MAXPNT,PRNT,REDO,REREAD
INTEGER EXIT,PRNT,REDO,REREAD

COMMON /ERROR/ IERROR,IWARN

COMMON /INIT/ CONFLO,MAXPA,MAXLR,MAXSE

COMMON /RD1/ ADCN,ADPH,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,CONF,CPNI,
+ CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS,
+ FAC(10),FCS,FSD,H,INC,K,M,MCRS,MXHS,
+ NCF,NDP,NSP,OST,PIUP,PHE,PPRS,PRODX,REFURB,RNC,
+ SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC,
+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR
INTEGER CADRE
REAL INC,K,MCRS,MXHS

COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),NCI(200),
+ MTBI(200),N,PA(200),QPA(200),QR(200),RM(200),
+ UC(200)
INTEGER QPA,QR
REAL MTBI

COMMON /RD5/ BCMH(200),CBCMH(200),CMH(200),CRTS1(200),CRTS2(200),
+ DBCMH(200),DMH(200),DRTS1(200),DRTS2(200),
+ DRTS3(200),PPR(200),IMH(200),INO15(200),N15,
+ PAMH(200),RIP(200),RL(200),RMH(200),SMH(200),SRTS1(200),
+ WOR1(200),WOR2(200),WOR3(200)
REAL IMH
INTEGER RL

2 FORMAT (/44H INPUT ERROR: ILLEGAL REPAIR LEVEL HAS BEEN/
+ 17X,25HSPECIFIED FOR LRU NUMBER ,I3,1H.)
3 FORMAT (/5CH INPUT ERROR: FOR RL OF 1, SRTS1 + CRTS1 + WOR1 +,
+ 6H DRTS1/17X,36HDOES NOT ADD UP TO 1 FOR LRU NUMBER ,I3,1H.)
4 FORMAT (/48H INPUT ERROR: FOR RL OF 2, CRTS2 + WOR2 + DRTS2/
+ 17X,36HDOES NOT ADD UP TO 1 FOR LRU NUMBER ,I3,1H.)
5 FORMAT (/40H INPUT ERROR: FOR RL OF 3, WOR3 + DRTS3/
+ 17X,36HDOES NOT ADD UP TO 1 FOR LRU NUMBER ,I3,1H.)
6 FORMAT (/50H INPUT ERROR: MTBI HAS BEEN INPUT AS 0.0 FOR LRU ,
+ 7HNUMBER ,I3,1H.)
8 FORMAT (1X/47H WARNING. INVALID SMTBI. DEFAULTS TO SMTBI=,

```

```

      *          P12.2)                                00000640
11  FORMAT (/44H WARNING:  CPMI = 0.  INFINITE CPMI ASSUMED.) 00000650
12  FORMAT (/44H WARNING:  CONF TOO SMALL.  DEFAULT TO CONF=,P10.5) 00000660
13  FORMAT (/43H WARNING:  NSP TOO LARGE.  DEFAULTS TO NSP=,I3) 00000670
C                                     00000680
      TEN1 = 0.                                         00000690
      IMAX = 0.                                         00000700
C                                     00000710
C*****00000720
C* LOOP ON LRUS TO PERFORM VARIOUS ERROR CHECKS:          *00000730
C* RL,SRTS1+CRTS1+WOR1+DRTS1,CRTS2+WOR2+DRTS2,WOR3+DRTS3,MTBI. *00000740
C*****00000750
C                                     00000760
      DO 32 I=1,N                                       00000770
      IF (QPA(I).EQ.0) GO TO 32                         00000780
C                                     00000790
C*****00000800
C* DEPENDING ON THE REPAIR LEVEL FOR THE CURRENT LRU, CONTROL IS *00000810
C* PASSED TO ONE OF FOUR SECTIONS OF CODE.  IN THE EVENT OF AN IL- *00000820
C* LEGAL REPAIR LEVEL (LESS THAN 0 OR GREATER THAN 3) AN ERROR MES- *00000830
C* SAGE IS PRINTED.                                     *00000840
C*****00000850
C                                     00000860
      IF (RL(I).EQ.0) GO TO 24                          00000870
      IF (RL(I).EQ.1) GO TO 21                          00000880
      IF (RL(I).EQ.2) GO TO 22                          00000890
      IF (RL(I).EQ.3) GO TO 23                          00000900
      WRITE (6,2) I                                     00000910
      IF (PRNT.NE.0) WRITE (7,2) I                     00000920
      IERROR=IERROR+1                                  00000930
      GO TO 24                                           00000940
C                                     00000950
C*****00000960
C* FOR RL=1, SRTS1(I), CRTS1(I), WOR1(I), AND DRTS1(I) MUST ADD UP *00000970
C* TO 1.  IF NOT, AN ERROR MESSAGE IS PRINTED.         *00000980
C*****00000990
C*                                     00001000
21  T1 = SRTS1(I) + CRTS1(I) + WOR1(I) + DRTS1(I)      00001010
      IF (RL(I).EQ.1.AND.T1.GE..99999.AND.T1.LE.1.00001) GO TO 24 00001020
      WRITE (6,3) I                                     00001030
      IF (PRNT.NE.0) WRITE (7,3) I                     00001040
      IERROR=IERROR+1                                  00001050
      GO TO 24                                           00001060
C                                     00001070
C*****00001080
C* FOR RL=2, CRTS2(I), WOR2(I), AND DRTS2(I) MUST ADD UP TO 1.  IF *00001090
C* NOT, AN ERROR MESSAGE IS PRINTED.                   *00001100
C*****00001110
C*                                     00001120
22  T2 = CRTS2(I) + WOR2(I) + DRTS2(I)                00001130
      IF (RL(I).EQ.2.AND.T2.GE..99999.AND.T2.LE.1.00001) GO TO 24 00001140
      WRITE (6,4) I                                     00001150
      IF (PRNT.NE.0) WRITE (7,4) I                     00001160
      IERROR=IERROR+1                                  00001170
      GO TO 24                                           00001180
C                                     00001190
C*****00001200
C* FOR RL=3, WOR3(I) AND DRTS3(I) MUST ADD UP TO 1.  IF NOT, AN ERROR *00001210
C* MESSAGE IS PRINTED.                                 *00001220
C*****00001230
C*                                     00001240
23  T3 = WOR3(I) + DRTS3(I)                            00001250
      IF (RL(I).EQ.3.AND.T3.GE..99999.AND.T3.LE.1.00001) GO TO 24 00001260

```


AD-A058 632

MITRE CORP BEDFORD MASS

SEEK IGLOO LIFE CYCLE COST MODEL. VOLUME III. MAINTENANCE MANUA--ETC(U)

F/G 9/2

JUL 78 J K FERRAILO

F19628-78-C-0001

UNCLASSIFIED

MTR-3577-VOL-3

ESO-TR-78-155-VOL-3

NL

2 OF 2

AD
A058632

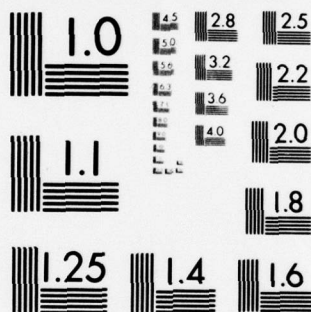


END

DATE
FILMED

11-78

DOC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```

        WRITE (6,5) I                                00001270
        IF (PRNT.NE.0) WRITE (7,5) I                  00001280
        IERROR=IERROR+1                                00001290
24      CONTINUE                                      00001300
C                                                00001310
C*****00001320
C* IF ANY MTBI(I) IS INPUT AS 0.0, AN ERROR MESSAGE IS PRINTED OUT, *00001330
C* AND SKIP THE FOLLOWING ERROR CHECK AND SUMMATION. *00001340
C*****00001350
C                                                00001360
        IF (MTBI(I).GT.0.001) GO TO 25                00001370
        WRITE (6,6) I                                00001380
        IF (PRNT.NE.0) WRITE (7,6) I                  00001390
        IERROR=IERROR+1                                00001400
        GO TO 32                                        00001410
25      CONTINUE                                      00001420
C                                                00001430
C*****00001440
C* SET TEM1 TO SUM (QPA(I)/MTBI(I),I=1,N). *00001450
C*****00001460
C                                                00001470
        TEM1 = TEM1 + (QPA(I) / MTBI(I))              00001480
32      CONTINUE                                      00001490
C                                                00001500
C*****00001510
C* END OF LOOP ON LRUS. NOW CHECK VARIOUS SCALARS FOR ERRORS: *00001520
C* SMTBI,CPHI,CONF,NSP *00001530
C*****00001540
C                                                00001550
        IF (SMTBI.GT.0.00001) TEM2 = (1./ SMTBI)      00001560
        IF (TEM2.GE.TEM1.AND.SMTBI.GT.0.00001) GO TO 33 00001570
        SMTBI = 1./ TEM1                                00001580
        WRITE(6,8) SMTBI                              00001590
        IF (PRNT.NE.0) WRITE(7,8) SMTBI               00001600
        IWARN = IWARN + 1                              00001610
33      CONTINUE                                      00001620
C                                                00001630
C                                                00001640
        IF (CPHI.GT.0.00001) GO TO 36                00001650
        WRITE(6,11)                                    00001660
        IF (PRNT.NE.0) WRITE(7,11)                    00001670
        IWARN = IWARN + 1                              00001680
36      CONTINUE                                      00001690
C                                                00001700
        IF (CONF.GE.CONFLO) GO TO 37                  00001710
        CONF = CONFLO                                  00001720
        WRITE(6,12) CONF                               00001730
        IF (PRNT.NE.C) WRITE(7,12) CONF               00001740
        IWARN = IWARN + 1                              00001750
37      CONTINUE                                      00001760
C                                                00001770
        IF (NSP.LE.3) GO TO 38                        00001780
        NSP = 3                                         00001790
        WRITE(6,13) NSP                                00001800
        IF (PRNT.NE.C) WRITE(7,13) NSP                00001810
        IWARN = IWARN + 1                              00001820
38      CONTINUE                                      00001830
C                                                00001840
        RETURN                                          00001850
        END                                            00001860
        SUBROUTINE AUXILI                             00001870
C                                                00001880
C*****00001890

```

C*	SUBROUTINE AUXIL1	*00001900
C*	THIS SUBROUTINE CALCULATES THE FOLLOWING AUXILIARY VARIABLES:	*00001910
C*	CCOND(I), COND(I), CRTS(I), DCOND(I), DRTS(I), SNRTS(I), SRTS(I), WR(I)	*00001920
C**	*****	*00001930
C	COMMON /RD4/ INO(200), LRNAME(200,24), LRPART(200,12), HCI(200),	00001940
+	MTBI(200), N, PA(200), QPA(200), QR(200), RM(200),	00001950
+	UC(200)	00001960
	INTEGER QPA, QR	00001970
	REAL MTBI	00001980
C	COMMON /RD5/ BCNH(200), CBCNH(200), CHH(200), CRTS1(200), CRTS2(200),	00002000
+	DBCNH(200), DHH(200), DRTS1(200), DRTS2(200),	00002010
+	DRTS3(200), FPR(200), INH(200), INO15(200), N15,	00002020
+	PAMH(200), RIP(200), RL(200), RMH(200), SHH(200), SRTS1(200),	00002030
+	WOR1(200), WOR2(200), WOR3(200)	00002040
	REAL INH	00002050
	INTEGER RL	00002060
C	COMMON /AUX1/ CCOND(200), COND(200), CRTS(200), DCOND(200),	00002070
+	DRTS(200), SNRTS(200), SRTS(200), WR(200)	00002080
C	DO 5 I=1, N	00002090
	IF (QPA(I).EQ.0) GO TO 5	00002100
	IF (RL(I).NE.0) GO TO 1	00002110
	COND(I) = 1.	00002120
	SRTS(I) = 0.	00002130
	CRTS(I) = 0.	00002140
	CCOND(I) = 0.	00002150
	DCOND(I) = 0.	00002160
	DRTS(I) = 0.	00002170
	GO TO 4	00002180
1	IF (RL(I).NE.1) GO TO 2	00002190
	COND(I) = 0.	00002200
	SRTS(I) = SRTS1(I)	00002210
	CRTS(I) = CRTS1(I)	00002220
	CCOND(I) = WOR1(I)	00002230
	DCOND(I) = 0.	00002240
	DRTS(I) = DRTS1(I)	00002250
	GO TO 4	00002260
2	IF (RL(I).NE.2) GO TO 3	00002270
	COND(I) = 0.	00002280
	SRTS(I) = 0.	00002290
	CRTS(I) = CRTS2(I)	00002300
	CCOND(I) = WOR2(I)	00002310
	DCOND(I) = 0.	00002320
	DRTS(I) = DRTS2(I)	00002330
	GO TO 4	00002340
3	CONTINUE	00002350
	COND(I) = 0.	00002360
	SRTS(I) = 0.	00002370
	CRTS(I) = 0.	00002380
	CCOND(I) = 0.	00002390
	DCOND(I) = WOR3(I)	00002400
	DRTS(I) = DRTS3(I)	00002410
4	CONTINUE	00002420
	SNRTS(I) = 1. - SRTS(I) - COND(I)	00002430
	WR(I) = COND(I) + CCOND(I) + DCOND(I)	00002440
5	CONTINUE	00002450
C	RETURN	00002460
	END	00002470
	SUBROUTINE AUXIL2	00002480
		00002490
		00002500
		00002510
		00002520


```

C
C*****00002530
C* SUBROUTINE AUXIL200002540
C* THIS SUBROUTINE CALCULATES THE FOLLOWING AUXILIARY VARIABLES:00002550
C* YPR(I),WPR(I)00002560
C*****00002570
C00002580
C00002590
COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,CONF,CPHI,00002600
+ CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS,00002610
+ PAC(10),PCS,PSD,H,INC,K,H,NCRS,MXHS,00002620
+ NCP,NDP,NSP,OST,PIUP,PNE,PPRS,PRODX,REFURB,RMC,00002630
+ SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC,00002640
+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR00002650
INTEGER CADRE00002660
REAL INC,K,NCRS,MXHS00002670
C00002680
COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),NCI(200),00002690
+ MTBI(200),N,PA(200),QPA(200),QR(200),RH(200),00002700
+ UC(200)00002710
INTEGER QPA,QR00002720
REAL MTBI00002730
C00002740
COMMON /RD5/ BCMH(200),CBCMH(200),CMH(200),CRTS1(200),CRTS2(200),00002750
+ DBCMH(200),DMH(200),DRTS1(200),DRTS2(200),00002760
+ DRTS3(200),FPR(200),INH(200),INO15(200),N15,00002770
+ PANH(200),RIP(200),RL(200),RMH(200),SMH(200),SRTS1(200),00002780
+ WOR1(200),WOR2(200),WOR3(200)00002790
REAL INH00002800
INTEGER RL00002810
C00002820
COMMON /AUX2/ YPR(200),WPR(200)00002830
C00002840
DO 5 I=1,N00002850
IF (QPA(I).EQ.0) GO TO 500002860
YPR(I) = (FLOAT(N) * YOH * K * FLOAT(QPA(I))00002870
+ * (1. - RIP(I))) / MTBI(I)00002880
WPR(I) = YPR(I) / 52.1800002890
5 CONTINUE00002900
C00002910
RETURN00002920
END00002930
SUBROUTINE AUXIL300002940
C00002950
C*****00002960
C* SUBROUTINE AUXIL300002970
C* THIS SUBROUTINE CALCULATES THE FOLLOWING AUXILIARY VARIABLES:00002980
C* CCLH(I),DCLH(I),DLH00002990
C*****00003000
C00003010
COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),NCI(200),00003020
+ MTBI(200),N,PA(200),QPA(200),QR(200),RH(200),00003030
+ UC(200)00003040
INTEGER QPA,QR00003050
REAL MTBI00003060
C00003070
COMMON /RD5/ BCMH(200),CBCMH(200),CMH(200),CRTS1(200),CRTS2(200),00003080
+ DBCMH(200),DMH(200),DRTS1(200),DRTS2(200),00003090
+ DRTS3(200),FPR(200),INH(200),INO15(200),N15,00003100
+ PANH(200),RIP(200),RL(200),RMH(200),SMH(200),SRTS1(200),00003110
+ WOR1(200),WOR2(200),WOR3(200)00003120
REAL INH00003130
INTEGER RL00003140
C00003150

```

```

COMMON /AUX1/ CCOND(200),COND(200),CRTS(200),DCOND(200),
+ DRTS(200),SNRTS(200),SRTS(200),WR(200)
C
COMMON /AUX2/ YPR(200),WPR(200)
C
COMMON /AUX3/ CCLH(200),DCLH(200),DLH
C
DLH = 0.
DO 5 I=1,N
  IF (QPA(I).EQ.0) GO TO 5
  CCLH(I) = YPR(I) * ((1.+ FPR(I)) * SNRTS(I) * CBCNH(I)
+ CRTS(I) * CNH(I) + FPR(I) * COND(I) * CBCNH(I))
  DCLH(I) = YPR(I) * ((DCOND(I) + DRTS(I)) * DBCNH(I) + DRTS(I)
+ DMH(I))
  DLH = DLH + DCLH(I)
5 CONTINUE
C
RETURN
END
SUBROUTINE AUXIL4
C
C*****
C* SUBROUTINE AUXIL4
C* THIS SUBROUTINE CALCULATES THE FOLLOWING AUXILIARY VARIABLES:
C* CAS(I),DAS(I),SAS(I)
C*****
C
COMMON /BD1/ ADCH,ADPH,B,CAA,CADRE,CCHP,CDR,CDWH,CLR,CONF,CPHI,
+ CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS,
+ FAC(10),FCS,FSD,H,INC,K,H,NCRS,NXHRS,
+ MCP,NDP,NSP,OST,PIUP,PHE,PPRS,PRCDX,REFURB,RNC,
+ SAA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC,
+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR
INTEGER CADRE
REAL INC,K,NCRS,NXHRS
C
COMMON /BD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200),
+ MTBI(200),N,PA(200),QPA(200),QR(200),RH(200),
+ UC(200)
INTEGER QPA,QR
REAL MTBI
C
COMMON /BD5/ BCHH(200),CBCNH(200),CNH(200),CRTS1(200),CRTS2(200),
+ DBCNH(200),DMH(200),DRTS1(200),DRTS2(200),
+ DRTS3(200),FPR(200),INH(200),INO15(200),N15,
+ PANH(200),RIP(200),RL(200),RHH(200),SHH(200),SRTS1(200),
+ WOR1(200),WOR2(200),WOR3(200)
REAL INH
INTEGER RL
C
COMMON /AUX1/ CCOND(200),COND(200),CRTS(200),DCOND(200),
+ DRTS(200),SNRTS(200),SRTS(200),WR(200)
C
COMMON /AUX2/ YPR(200),WPR(200)
C
COMMON /AUX4/ CAS(200),DAS(200),SAS(200)
C
DO 5 I=1,N
  IF (QPA(I).EQ.0) GO TO 5
  SAS(I) = (WPR(I) * (1.+ FPR(I)) * (SRTS(I) * SRCT
+ (1.- SRTS(I)) * SOSI)) / FLOAT(N)
  CAS(I) = WPR(I) * (FPR(I) * (1.- SRTS(I)) * CRCT
+ CRTS(I) * CRCT + (CCOND(I) + DRTS(I) + DCOND(I))

```

```

      * (OST + SOSI / 2.) + COND(I) * OST)
      CAS(I) = WPR(I) * DRTS(I) * DRCT
5    CONTINUE
C
      RETURN
      END
      SUBROUTINE AUXIL5
C
C*****
C*          SUBROUTINE AUXIL5
C* THIS SUBROUTINE CALCULATES THE FOLLOWING AUXILIARY VARIABLES:
C* DSTK(I), ECMT, MCF(I), STK(I), STK1(I), Z
C*****
C
      COMMON /CNTL/ EXIT, ITER, MAXPMT, PRNT, REDO, REREAD
      INTEGER EXIT, PRNT, REDO, REREAD
C
      COMMON /INIT/ CONFLO, MAXFA, MAXLR, MAXSE
C
      COMMON /RD1/ ADCH, ADPH, B, CAA, CADRE, CCMP, CDR, CDWH, CLR, CONF, CPNI,
      *             CPHP, CPWT, CRCT, DAA, DLR, DECT, DS,
      *             FAC(10), PCS, FSD, H, INC, K, M, MCRS, MIHRS,
      *             NCP, NDP, MSP, OST, PIUP, PHE, PPRS, PRODX, REFURB, RMC,
      *             SA, SAA, SMTBI, SOSI, SPRS, SRCT, SW, SWFAC,
      *             TC, TCCAD, TE, TR, TW, TWCAD, VAL, XUC, YOH, YSCAD, YSLR
      INTEGER CADRE
      REAL INC, K, MCRS, MIHRS
C
      COMMON /RD4/ INO(200), LRNAME(200,24), LRPART(200,12), MCI(200),
      *             MTBI(200), N, PA(200), QPA(200), QR(200), RH(200),
      *             UC(200)
      INTEGER QPA, QR
      REAL MTBI
C
      COMMON /RD5/ BCCH(200), CBCCH(200), CHH(200), CRTS1(200), CRTS2(200),
      *             DBCMH(200), DMH(200), DRTS1(200), DRTS2(200),
      *             DRTS3(200), FPR(200), IMH(200), INO15(200), M15,
      *             PAHH(200), RIP(200), RL(200), RHH(200), SHH(200), SRTS1(200),
      *             WOR1(200), WOR2(200), WOR3(200)
      REAL IMH
      INTEGER RL
C
      COMMON /AUX2/ YPR(200), WPR(200)
C
      COMMON /AUX4/ CAS(200), DAS(200), SAS(200)
C
      COMMON /AUX5/ DSTK(200), ECMT, MCF(200), STK(200), STK1(200), Z
      INTEGER STK, STK1
C
1    FORMAT (1X//49H ERROR ENCOUNTERED IN COMPUTATION OF STK1 FOR LRJ,
      * I4,1H./49H THE SERIES DID NOT CONVERGE. PROGRAM CONTINUES.)
C
      PIVAL = 1. / SQRT(2. * 3.1415926)
      Z = 1.
      ZC = 0.025
      AREA = CONFLO
      IF (CONF.LE.AREA) GO TO 5
3    CONTINUE
      DO 4 MH=1,80
        Z = Z + ZC
        AREA = AREA + (PIVAL * (EXP((Z**2)/(-2.))
      *             + EXP(((Z-ZC)**2)/(-2.)) * ZC) / 2.
      IF (AREA.GE.CONF) GO TO 5

```


4	CONTINUE	00004420
C		00004430
5	ECNT = (FLOAT(M) * YOH * K * (1.- DS)) / SMTBI	00004440
	DO 8 I=1,N	00004450
	IF (QPA(I).EQ.0) GO TO 8	00004460
	T1 = EXP(-1.* SAS(I))	00004470
	TFM1 = T1	00004480
	DO 6 M1=1,200	00004490
	IF (TEM1.GE.CONF) GO TO 7	00004500
	T1 = (SAS(I) * T1) / FLOAT(M1)	00004510
	TEM1 = TEM1 + T1	00004520
6	CONTINUE	00004530
	WRITE(6,1) I	00004540
	IF (PRNT.NE.0) WRITE(7,1) I	00004550
7	STK1(I) = MAX0(C, (M1 - QR(I)) - 1)	00004560
	STK(I) = MAX0(STK1(I), MCI(I) - 1)	00004570
	FLMCI = FLOAT(MCI(I))	00004580
	IF (STK(I).GT.0) DSTK(I) = 1. + Z / (2. * SQRT(SAS(I)))	00004590
	IF (STK(I).EQ.0) DSTK(I) = 1. / ((SQRT(FLOAT(QR(I)) +	00004600
+	(FLMCI / 2.) * (FLMCI - 1.)	00004610
+	+ 0.5 + (Z**2) / 4.) - (Z / 2.)) ** 2)	00004620
	K2 = STK(I) + QR(I)	00004630
	MCF(I) = C	00004640
	IF (MCI(I).EQ.1.AND.K2.EQ.0) MCF(I) = 1	00004650
	ECNT = ECNT + FLOAT(MCF(I)) * YPR(I)	00004660
8	CONTINUE	00004670
C		00004680
	RETURN	00004690
	END	00004700


```

SUBROUTINE COST1
C*****
C* SUBROUTINE COST1
C* THIS SUBROUTINE CALCULATES COST ELEMENT 1, ACQUISITION AND
C* DEVELOPMENT.
C*****
COMMON /C1/ C1
C
COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,CONF,CPHI,
+ CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS,
+ PAC(10),PCS,PSD,H,INC,K,H,HCRS,HXHRS,
+ NCP,NDP,NSP,OST,PIUP,PNE,PPRS,PRODX,REFURB,RNC,
+ SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC,
+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR
INTEGER CADRE
REAL INC,K,HCRS,HXHRS
C1 = VAL + PSD + (XUC * PNE) + REFURB + SW + PRODX
RETURN
END
SUBROUTINE COST2
C*****
C* SUBROUTINE COST2
C* THIS SUBROUTINE CALCULATES COST ELEMENT 2, MAINTENANCE (LABOR AND
C* TRANSPORTATION).
C*****
COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,CONF,CPHI,
+ CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS,
+ PAC(10),PCS,PSD,H,INC,K,H,HCRS,HXHRS,
+ NCP,NDP,NSP,OST,PIUP,PNE,PPRS,PRODX,REFURB,RNC,
+ SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC,
+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR
INTEGER CADRE
REAL INC,K,HCRS,HXHRS
COMMON /RD4/ INO(200),LNNAME(200,24),LRPART(200,12),NCI(200),
+ MTBI(200),N,PA(200),QPA(200),QR(200),RH(200),
+ UC(200)
INTEGER QPA,QR
REAL MTBI
COMMON /AUX2/ YPR(200),WPR(200)
COMMON /AUX3/ CCLH(200),DCLH(200),DLH
COMMON /AUX5/ DSTK(200),ECHT,MCF(200),STK(200),STK1(200),Z
INTEGER STK,STK1
COMMON /C2/ C2,C2C,C2D,C2L(200),C2S
C2S = FLOAT(H) * FLOAT(NSP) * YSLR * PIUP
IF (CPHI.LE.0.00001) C2C = 0.
IF (CPHI.GT.0.00001) C2C = FLOAT(H)
+ PIUP * YOH * (CPMP * ADPH * CDR + H) / CPHI
C2C = C2C + ((PIUP * FLOAT(H) * YOH * K * (1.- DS)) / SMTBI)
+ (CCMP * ADCH * CDR + H)
C2D = 0.
C2 = C2S + C2C

```

```

DO 5 I=1,N
  IF (QPA(I).EQ.0) GO TO 5
  TEM2 = PIUP * CCLH(I) * CLR + PIUP * FLOAT(NCF(I)) * YPR(I)
  * (CCMP * ADCH * CDR + H)
  TEM3 = PIUP * DCLH(I) * DLR
  C2L(I) = TEM2 + TEM3
  C2C = C2C + TEM2
  C2D = C2D + TEM3
  C2 = C2 + C2L(I)
5 CONTINUE
C
  RETURN
END
SUBROUTINE COST3
C*****
C* SUBROUTINE COST3
C* THIS SUBROUTINE CALCULATES COST ELEMENT 3, INVESTMENT SPARES.
C*****
C COMMON /C3/ C3,C3C,C3D,C3L(200),C3S
C
C COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,CONF,CPHI,
+ CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS,
+ FAC(10),FCS,PSD,H,INC,K,H,NCRS,NHRS,
+ MCP,NDP,NSP,OST,PIUP,PHE,PPRS,PRCDX,REFURB,RNC,
+ SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC,
+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR
C INTEGER CADRE
C REAL INC,K,NCRS,NHRS
C
C COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200),
+ MTBI(200),N,PA(200),QPA(200),QR(200),RH(200),
+ UC(200)
C INTEGER QPA,QR
C REAL MTBI
C
C COMMON /AUX4/ CAS(200),DAS(200),SAS(200)
C
C COMMON /AUX5/ DSTK(200),ECMT,NCF(200),STK(200),STK1(200),Z
C INTEGER STK,STK1
C
C3 = 0.
C3S = 0.
C3C = 0.
C3D = 0.
DO 5 I=1,N
  IF (QPA(I).EQ.0) GO TO 5
  TEM1 = FLOAT(H) * FLOAT(STK(I)) * XUC * UC(I)
  TEM2 = AINT(CAS(I) + B * SQRT(CAS(I)) + 0.5) * XUC * UC(I)
  TEM3 = AINT(DAS(I) + B * SQRT(DAS(I)) + 0.5) * XUC * UC(I)
  C3L(I) = TEM1 + TEM2 + TEM3
  C3 = C3 + C3L(I)
  C3S = C3S + TEM1
  C3C = C3C + TEM2
  C3D = C3D + TEM3
5 CONTINUE
C
  RETURN
END
SUBROUTINE COST4
C*****

```

```

C*                SUBROUTINE COST4                *00001270
C*  THIS SUBROUTINE CALCULATES COST ELEMENT 4, REPLENISHMENT SPARES.  *00001280
C*****00001290
C                COMMON /C4/ C4,C4I,C4L(200),C4R                00001300
C                COMMON /RD1/ ADCN,ADPH,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,COMP,CPHI, 00001310
+                CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS,                00001320
+                PAC(10),FCS,PSD,H,INC,K,H,MCRS,MXHS,                00001330
+                NCP,NDP,NSP,OST,PIUP,PNE,PPRS,PRCDX,REFURB,RMC, 00001340
+                SA,SAA,SHTBI,SOSI,SPRS,SRCT,SW,SWFAC,                00001350
+                TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR 00001360
+                INTEGER CADRE                00001370
+                REAL INC,K,MCRS,MXHS                00001380
C                COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200), 00001390
+                MTBI(200),N,PA(200),QPA(200),QR(200),RM(200), 00001400
+                UC(200)                00001410
+                INTEGER QPA,QR                00001420
+                REAL MTBI                00001430
C                COMMON /AUX1/ CCOND(200),COND(200),CRTS(200),DCOND(200), 00001440
+                DRTS(200),SWRTS(200),SRTS(200),WR(200)                00001450
C                COMMON /AUX2/ YPR(200),WPR(200)                00001460
C                C4 = 0.                00001470
C                C4I = 0.                00001480
C                C4R = 0.                00001490
C                DO 5 I=1,N                00001500
+                IF (QPA(I).EQ.0) GO TO 5                00001510
+                TEM1 = YPR(I) * (WR(I) + (1.-WR(I)) * RM(I)) * XUC * UC(I) 00001520
+                TEM2 = (PIUP - 1.) * TEM1                00001530
+                C4L(I) = TEM1 + TEM2                00001540
+                C4 = C4 + C4L(I)                00001550
+                C4I = C4I + TEM1                00001560
+                C4R = C4R + TEM2                00001570
5                CONTINUE                00001580
C                RETURN                00001590
C                END                00001600
C                SUBROUTINE COST5                00001610
C                COMMON /C5/ C5                00001620
C                COMMON /RD1/ ADCN,ADPH,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,COMP,CPHI, 00001630
+                CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS,                00001640
+                PAC(10),FCS,PSD,H,INC,K,H,MCRS,MXHS,                00001650
+                NCP,NDP,NSP,OST,PIUP,PNE,PPRS,PRCDX,REFURB,RMC, 00001660
+                SA,SAA,SHTBI,SOSI,SPRS,SRCT,SW,SWFAC,                00001670
+                TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR 00001680
+                INTEGER CADRE                00001690
+                REAL INC,K,MCRS,MXHS                00001700
C                C5 = FLOAT(N) * PIUP * ((PPRS + SPRS) * YOH * FCS + MCRS) 00001710
C                RETURN                00001720
C                END                00001730
C*****00001740
C*                SUBROUTINE COST5                *00001710
C*  THIS SUBROUTINE CALCULATES COST ELEMENT 5, CONSUMABLES.  *00001720
C*****00001730
C                COMMON /C5/ C5                00001740
C                COMMON /RD1/ ADCN,ADPH,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,COMP,CPHI, 00001750
+                CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS,                00001760
+                PAC(10),FCS,PSD,H,INC,K,H,MCRS,MXHS,                00001770
+                NCP,NDP,NSP,OST,PIUP,PNE,PPRS,PRCDX,REFURB,RMC, 00001780
+                SA,SAA,SHTBI,SOSI,SPRS,SRCT,SW,SWFAC,                00001790
+                TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR 00001800
+                INTEGER CADRE                00001810
+                REAL INC,K,MCRS,MXHS                00001820
C                C5 = FLOAT(N) * PIUP * ((PPRS + SPRS) * YOH * FCS + MCRS) 00001830
C                RETURN                00001840
C                END                00001850

```



```

SUBROUTINE COST6
C
C*****00001900
C*****00001910
C*****00001920
C* SUBROUTINE COST6 *00001930
C* THIS SUBROUTINE CALCULATES COST ELEMENT 6, SUPPORT EQUIPMENT. *00001940
C*****00001950
C
COMMON /C6/ C6, C6C, C6D, C6I, C6R, C6S
C
COMMON /RD1/ ADCM,ADPH,B,CAA,CADRE,CCHP,CDR,CDWH,CLR,CONF,CPHI,
+ CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS,
+ FAC(10),PCS,PSD,H,INC,K,H,MCRS,MHRS,
+ NCP,NDP,NSP,OST,PIUP,PNE,PPRS,PRODX,REFURB,RHC,
+ SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC,
+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,YUC,YOH,YSCAD,YSLR
INTEGER CADRE
REAL INC,K,MCRS,MHRS
C
COMMON /RD2/ A,COC(150),COD(150),COS(150),INOSE(150),NSEC(150),
+ NSED(150),NSES(150),SEC(150),SENAME(150,24)
REAL NSEC,NSED,NSES
INTEGER A
C
C6 = 0.
C6I = 0.
C6R = 0.
C6S = 0.
C6C = 0.
C6D = 0.
IF (A.EQ.0) GO TO 6
DO 5 L=1,A
TOT = NSES(L) + NSEC(L) + NSED(L)
IF (TOT.LE.0.0001) GO TO 5
TEM1 = FLOAT(N) * NSES(L) * SEC(L)
TEM2 = TEM1 * PIUP * COS(L)
TEM3 = NSEC(L) * SEC(L)
TEM4 = TEM3 * PIUP * COC(L)
TEM5 = NSED(L) * SEC(L)
TEM6 = TEM5 * PIUP * COD(L)
C6 = C6 + TEM1 + TEM2 + TEM3 + TEM4 + TEM5 + TEM6
C6I = C6I + TEM1 + TEM3 + TEM5
C6R = C6R + TEM2 + TEM4 + TEM6
C6S = C6S + TEM1 + TEM2
C6C = C6C + TEM3 + TEM4
C6D = C6D + TEM5 + TEM6
5 CONTINUE
6 CONTINUE
C
RETURN
END
SUBROUTINE COST7
C
C*****00002410
C*****00002420
C* SUBROUTINE COST7 *00002430
C* THIS SUBROUTINE CALCULATES COST ELEMENT 7, FACILITIES. *00002440
C*****00002450
C
COMMON /C7/ C7
C
COMMON /RD1/ ADCM,ADPH,B,CAA,CADRE,CCHP,CDR,CDWH,CLR,CONF,CPHI,
+ CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS,
+ FAC(10),PCS,PSD,H,INC,K,H,MCRS,MHRS,
+ NCP,NDP,NSP,OST,PIUP,PNE,PPRS,PRCDX,REFURB,RHC,

```



```

+          SA,SAA,SHTBI,SOSI,SPRS,SRCT,SW,SWFAC,          00002530
+          TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR  00002540
INTEGER CADRE      00002550
REAL INC,K,MCRS,MXHRS 00002560
C          00002570
C7 = 0.          00002580
DO 5 LL=1,10     00002590
  C7 = C7 + FLOAT(M) * FAC(LL) 00002600
5 CONTINUE      00002610
C          00002620
RETURN          00002630
END            00002640
SUBROUTINE COST8 00002650
C          00002660
C*****00002670
C*          SUBROUTINE COST8 *00002680
C* THIS SUBROUTINE CALCULATES COST ELEMENT 8, TRAINING COSTS. *00002690
C*****00002700
C          00002710
COMMON /C8/ C8,C8C,C8D,C8I,C8R,C8S 00002720
C          00002730
COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCHP,CDR,CDWH,CLR,CONF,CPI, 00002740
+          CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS, 00002750
+          FAC(10),PCS,PSD,H,INC,K,M,MCRS,MXHRS, 00002760
+          NCP,NDP,NSP,OST,PIUP,PHE,PPRS,PRODX,REFURB,RMC, 00002770
+          SA,SAA,SHTBI,SOSI,SPRS,SRCT,SW,SWFAC, 00002780
+          TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR 00002790
INTEGER CADRE      00002800
REAL INC,K,MCRS,MXHRS 00002810
C          00002820
TEM1 = FLOAT(M) * FLOAT(NSP) * TW * TC 00002830
TEM2 = TEM1 * (PIUP - 1.) * TR 00002840
TEM3 = FLOAT(CADRE) * TWCAD * TCCAD + TE + FLOAT(NCP) * TW * TC 00002850
TEM4 = FLOAT(CADRE) * YSCAD * PIUP + FLOAT(NCP) * (PIUP - 1.) 00002860
+          * TR * TW * TC 00002870
TEM5 = FLOAT(NDP) * TW * TC 00002880
TEM6 = TEM5 * (PIUP - 1.) * TR 00002890
C8 = TEM1 + TEM2 + TEM3 + TEM4 + TEM5 + TEM6 00002900
C8I = TEM1 + TEM3 + TEM5 00002910
C8R = TEM2 + TEM4 + TEM6 00002920
C8S = TEM1 + TEM2 00002930
C8C = TEM3 + TEM4 00002940
C8D = TEM5 + TEM6 00002950
C          00002960
RETURN          00002970
END            00002980
SUBROUTINE COST9 00002990
C          00003000
C*****00003010
C*          SUBROUTINE COST9 *00003020
C* THIS SUBROUTINE CALCULATES COST ELEMENT 9, INVENTORY MANAGEMENT. *00003030
C*****00003040
C          00003050
COMMON /C9/ C9,C9C,C9D,C9I,C9L(200),C9R,C9S 00003060
C          00003070
COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCHP,CDR,CDWH,CLR,CONF,CPI, 00003080
+          CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS, 00003090
+          FAC(10),PCS,PSD,H,INC,K,M,MCRS,MXHRS, 00003100
+          NCP,NDP,NSP,OST,PIUP,PHE,PPRS,PRODX,REFURB,RMC, 00003110
+          SA,SAA,SHTBI,SOSI,SPRS,SRCT,SW,SWFAC, 00003120
+          TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR 00003130
INTEGER CADRE      00003140
REAL INC,K,MCRS,MXHRS 00003150

```

C	COMMON /RD4/ INO(200), LRNAME(200,24), LRPART(200,12), HCI(200),	00003160
	+ MTBI(200), N, PA(200), QPA(200), QR(200), RM(200),	00003170
	+ UC(200)	00003180
	INTEGER QPA,QR	00003190
	REAL MTBI	00003200
C	COMMON /AUX1/ CCOND(200), COND(200), CRTS(200), DCOND(200),	00003210
	+ DRTS(200), SNRTS(200), SRTS(200), WR(200)	00003220
C	C9 = 0.	00003230
	C9I = 0.	00003240
	C9R = 0.	00003250
	C9S = 0.	00003260
	C9C = 0.	00003270
	C9D = 0.	00003280
	DO 5 I=1,N	00003290
	IF (QPA(I).EQ.0) GO TO 5	00003300
	IF (SRTS(I).LE.0.00001) U1 = 0.	00003310
	IF (SRTS(I).GT.0.00001) U1 = 1.	00003320
	IF (CRTS(I).LE.0.00001) U2 = 0.	00003330
	IF (CRTS(I).GT.0.00001) U2 = 1.	00003340
	IF (COND(I).GE.0.99999) U3 = 0.	00003350
	IF (COND(I).LT.0.99999) U3 = 1.	00003360
	TEM2 = FLOAT(N) * SA * PIUP * (1. + U1 * PA(I))	00003370
	TEM4 = SA * PIUP * (1. + U2 * PA(I))	00003380
	TEM5 = INC * (1. + U3 * PA(I))	00003390
	TEM6 = PIUP * RMC * (1. + U3 * PA(I))	00003400
	C9L(I) = TEM2 + TEM4 + TEM5 + TEM6	00003410
	C9 = C9 + C9L(I)	00003420
	C9I = C9I + TEM5	00003430
	C9R = C9R + TEM2 + TEM4 + TEM6	00003440
	C9S = C9S + TEM2	00003450
	C9C = C9C + TEM4	00003460
	C9D = C9D + TEM5 + TEM6	00003470
5	CONTINUE	00003480
C	RETURN	00003490
	END	00003500
	SUBROUTINE COST10	00003510
C	*****	00003520
	SUBROUTINE COST10	00003530
C*	THIS SUBROUTINE CALCULATES COST ELEMENT 10, SOFTWARE MAINTENANCE.	00003540
C*****	*****	00003550
C	COMMON /C10/ C10, C10HW(10), C10I, C10R, C10SW(10)	00003560
C	COMMON /RD1/ ADCN,ADPH,B,CAA,CADRE,CCNP,CDR,CDWH,CLR,CONF,CPHI,	00003570
	+ CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS,	00003580
	+ PAC(10),PCS,PSD,H,INC,K,M,MCRS,MXHS,	00003590
	+ MCP,NDP,NSP,OST,PIUP,PME,PPRS,PRODX,REFURB,RMC,	00003600
	+ SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWPAC,	00003610
	+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,YUC,YOH,YSCAD,YSLR	00003620
	INTEGER CADRE	00003630
	REAL INC,K,MCRS,MXHS	00003640
C	COMMON /RD3/ ENYR(10),PA,PANAME(10,24),HWCF(10),INOPA(10),MMH(10),	00003650
	+ SIZE(10),SMI(10),SWCF(10),SWPIX(10),SWVAR(10),	00003660
	+ WEIGHT(10)	00003670
	INTEGER PA	00003680
	REAL MMH	00003690
C		00003700
		00003710
		00003720
		00003730
		00003740
		00003750
		00003760
		00003770
		00003780

	C10I = SWFAC	00003790
	C10R = 0.	00003800
C		00003810
	DO 5 J=1,PA	00003820
	C10HW(J) = PIUP * ENYR(J) * WEIGHT(J) * HWCF(J) * CPWT	00003830
	C10SW(J) = PIUP * ENYR(J) * (SIZE(J) * SWCF(J) * SWVAR(J)	00003840
	+ SWFIX(J))	00003850
	C10R = C10R + C10HW(J) + C10SW(J)	00003860
5	CONTINUE	00003870
6	C10 = C10I + C10R	00003880
C		00003890
	RETURN	00003900
	END	00003910

```

C          SUBROUTINE NONCOS                                00000010
C          00000020
C*****00000030
C*          SUBROUTINE NONCOS                                00000040
C* THIS SUBROUTINE CALCULATES CERTAIN NON-COST VARIABLES (THE 00000050
C* VARIABLES ARE CALLED NON-COST BECAUSE THEIR VALUES DO NOT ENTER 00000060
C* INTO THE COST ELEMENT SUMMATIONS) THAT DEAL WITH MAINTENANCE 00000070
C* MAN-HOURS. SOME OF THESE VARIABLES (SMNH,SPNH,STNH,CLH) WILL BE 00000080
C* CHECKED AGAINST CONSTRAINTS IN SUBROUTINE OTAB1 (DLH, 00000090
C* CALCULATED IN AUXIL3, WILL ALSO BE CHECKED AGAINST CONSTRAINTS IN 00000100
C* OTAB1). IF THE CONSTRAINTS ARE NOT MET, OTAB1 WILL PRINT ERROR 00000110
C* MESSAGES. NONCOS ALSO CALCULATES VARIABLE PHPA(J) (PREVENTIVE 00000120
C* MAINTENANCE MAN-HOURS FOR FUNCTIONAL AREA J), WHICH WILL BE 00000130
C* PRINTED OUT IN OTAB4. 00000140
C*****00000150
C          00000160
C          COMMON /NCOS/ CLH,PHPA(10),SMNH,SPNH,STNH 00000170
C          00000180
C          COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,CONF,CPHI, 00000190
C          + CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS, 00000200
C          + FAC(10),PCS,PSD,H,INC,K,H,MCRS,MXHS, 00000210
C          + NCP,NDP,NSP,OST,PIUP,PME,PPRS,PRODX,REFURB,RMC, 00000220
C          + SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC, 00000230
C          + TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR 00000240
C          INTEGER CADRE 00000250
C          REAL INC,K,MCRS,MXHS 00000260
C          00000270
C          COMMON /RD3/ ENYR(10),FA,PANAME(10,24),HWCF(10),INOFA(10),MMH(10), 00000280
C          + SIZE(10),SMI(10),SWCF(10),SWPIX(10),SWVAR(10), 00000290
C          + WEIGHT(10) 00000300
C          REAL MMH 00000310
C          INTEGER FA 00000320
C          00000330
C          COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200), 00000340
C          + MTBI(200),N,PA(200),QPA(200),QR(200),RM(200), 00000350
C          + UC(200) 00000360
C          INTEGER QPA,QR 00000370
C          REAL MTBI 00000380
C          00000390
C          COMMON /RD5/ BCMH(200),CBCMH(200),CMH(200),CRTS1(200),CRTS2(200), 00000400
C          + DBCMH(200),DHH(200),DRTS1(200),DRTS2(200), 00000410
C          + DRTS3(200),PPR(200),IMH(200),INO15(200),N15, 00000420
C          + PAMH(200),RIP(200),RL(200),RMH(200),SMH(200),SRTS1(200), 00000430
C          + WOR1(200),WOR2(200),WOR3(200) 00000440
C          REAL IMH 00000450
C          INTEGER RL 00000460
C          00000470
C          COMMON /AUX1/ CCOND(200),COND(200),CRTS(200),DCOND(200), 00000480
C          + DRTS(200),SNRTS(200),SRTS(200),WR(200) 00000490
C          00000500
C          COMMON /AUX2/ YPR(200),WPR(200) 00000510
C          00000520
C          COMMON /AUX3/ CCLH(200),DCLH(200),DLH 00000530
C          00000540
C          COMMON /AUX5/ DSTK(200),ECMT,MCP(200),STK(200),STK1(200),Z 00000550
C          INTEGER STK,STK1 00000560
C          00000570
C          IF (CPMI.LE.0.00001) CLH = ECMT * CCMP * ADCH * CDWH 00000580
C          IF (CPMI.GT.0.00001) CLH = (FLOAT(N) * YOH * CPMP * ADPH 00000590
C          + * CDWH) / CPMI + ECMT * CCMP * ADCH * CDWH 00000600
C          SMNH = 0. 00000610
C          SPNH = 0. 00000620
C          00000630

```



```

IF (PA.EQ.0) GO TO 2                                00000640
DO 1 J=1,PA                                          00000650
  PHPA(J) = 0.                                       00000660
  IF (SMI(J).GT.0.00001) PHPA(J) = (YOH * MMH(J)) / SMI(J) 00000670
  SMHH = SMHH + PHPA(J)                             00000680
1 CONTINUE                                           00000690
2 CONTINUE                                           00000700
C                                                     00000710
DO 3 I=1,N                                           00000720
  IF (QPA(I).EQ.0) GO TO 3                          00000730
  SMHH = SMHH + (YOH * K * FLOAT(QPA(I)) * ((1.+FPR(I)) * PAMH(I) 00000740
    + RIP(I) * IMH(I) + (1.-RIP(I)) * (1.+FPR(I)) * RMH(I))) / MTBI(I) 00000750
  SPMH = SPMH + (YPR(I) / FLOAT(N)) * ((1.+FPR(I)) * BCMH(I) 00000760
    + SRTS(I) * SHH(I))                             00000770
  CLH = CLH + CCLH(I)                               00000780
3 CONTINUE                                           00000790
C                                                     00000800
STHH = SMHH + SPMH                                 00000810
C                                                     00000820
RETURN                                              00000830
END                                                00000840
SUBROUTINE DXUC                                     00000850
C                                                     00000860
C*****00000870
C* SUBROUTINE DXUC                                *00000880
C* THIS SUBROUTINE CALCULATES THE DERIVATIVE OF LIFE CYCLE COST WITH *00000890
C* RESPECT TO XUC (GOVERNMENT SUPPLIED LRU ESCALATION FACTOR(GLOBAL)) *00000900
C*****00000910
COMMON /TDXUC/ TDXUC                                00000920
C                                                     00000930
C* COMMON /RD1/ ADCH,ADPM,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,CONF,CPHI, 00000940
C* CPMP,CPMT,CRCT,DAA,DLR,DRCT,DS, 00000950
C* PAC(10),PCS,FSD,H,INC,K,H,MCRS,MXHS, 00000960
C* NCP,NDP,NSP,OST,PIUP,PME,PPRS,PRODX,REFURB,RMC, 00000970
C* SA,SAI,SMTBI,SOSI,SPRS,SRCT,SW,SWPAC, 00000980
C* TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR 00000990
C* INTEGER CADRE 00001000
C* REAL INC,K,MCRS,MXHS 00001010
C                                                     00001020
COMMON /SENS/ FINC,LDCMH,LDCRTS,LDDMH,LDDRTS,LDERV,LDFPR,LDFR, 00001030
C* LDMCI,LDRM,LDSRTS,LDWOR,LDOC 00001040
C                                                     00001050
COMMON /C3/ C3,C3C,C3D,C3L(200),C3S 00001060
C                                                     00001070
COMMON /C4/ C4,C4I,C4L(200),C4R 00001080
C                                                     00001090
TDXUC = FINC * (PME * XUC + C3 + C4) 00001100
C                                                     00001110
RETURN 00001120
END 00001130
SUBROUTINE DFR 00001140
C                                                     00001150
C*****00001160
C* SUBROUTINE DFR                                *00001170
C* THIS SUBROUTINE CALCULATES THE DERIVATIVE OF LIFE CYCLE COST WITH *00001180
C* RESPECT TO FR = 1./ MTBI(I) (FAILURE RATE). *00001190
C* IF LDFR > 0, THEN *00001200
C* AFTER ALL DERIVATIVES HAVE BEEN CALCULATED, THE SUBROUTINE CALLS *00001210
C* SUBROUTINE TDSORT TO 'BUBBLE UP' TO THE TOP OF ARRAY IDFR THE *00001220
C* INDEX NUMBERS OF THE LDFR LARGEST DERIVATIVES. *00001230
C*****00001240

```

C	COMMON /TDFR/ IDFR(200),TDFR(200)	00001270
C	COMMON /CNTL/ EXIT,ITER,MAXPMT,PRNT,REDO,REREAD	00001280
	INTEGER EXIT,PRNT,REDO,REREAD	00001290
C	COMMON /RD1/ ADCN,ADPN,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,CONF,CPHI,	00001300
	+ CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS,	00001310
	+ PAC(10),PCS,PSD,H,INC,K,M,MCRS,MXHS,	00001320
	+ NCP,NDP,NSP,OST,PIUP,PME,PPRS,PRODX,REFURB,RMC,	00001330
	+ SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC,	00001340
	+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR	00001350
	INTEGER CADRE	00001360
	REAL INC,K,MCRS,MXHS	00001370
C	COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200),	00001380
	+ MTBI(200),N,PA(200),QPA(200),QR(200),RM(200),	00001390
	+ UC(200)	00001400
	INTEGER QPA,QR	00001410
	REAL MTBI	00001420
C	COMMON /RD5/ BCMH(200),CBCMH(200),CMH(200),CRTS1(200),CRTS2(200),	00001430
	+ DBCM(200),DMH(200),DRTS1(200),DRTS2(200),	00001440
	+ DRTS3(200),FPR(200),IMH(200),INO15(200),N15,	00001450
	+ PAMH(200),RIP(200),RL(200),RMH(200),SMH(200),SRTS1(200),	00001460
	+ WOR1(200),WOR2(200),WOR3(200)	00001470
	REAL IMH	00001480
	INTEGER RL	00001490
C	COMMON /SENS/ PINC,LDCMH,LDCRTS,LDDMH,LDDRTS,LDERV,LDPPR,LDPR,	00001500
	+ LDMCI,LDRN,LDSRTS,LDWOR,LDUC	00001510
C	COMMON /AUX1/ CCOND(200),COND(200),CRTS(200),DCOND(200),	00001520
	+ DRTS(200),SNRTS(200),SRTS(200),WR(200)	00001530
C	COMMON /AUX2/ YPR(200),WPR(200)	00001540
C	COMMON /AUX3/ CCLH(200),DCLH(200),DLH	00001550
C	COMMON /AUX4/ CAS(200),DAS(200),SAS(200)	00001560
C	COMMON /AUX5/ DSTK(200),ECHT,MCP(200),STK(200),STK1(200),Z	00001570
	INTEGER STK,STK1	00001580
C	DO 5 I=1,N	00001590
	IF (QPA(I).NE.0) GO TO 1	00001600
	TDFR(I) = 0.	00001610
	GO TO 4	00001620
1	TDFR(I) = DSTK(I) * FLOAT(M) * PINC * SAS(I) * XUC * UC(I)	00001630
	+ (PINC * CAS(I) + B * (SQRT((1.+PINC) * CAS(I))	00001640
	- SQRT(CAS(I)))) * XUC * UC(I)	00001650
	+ (PINC * DAS(I) + B * (SQRT((1.+PINC) * DAS(I))	00001660
	- SQRT(DAS(I)))) * XUC * UC(I)	00001670
	+ PINC * PIUP * YPR(I) * (WR(I) + (1. - WR(I))	00001680
	* RM(I)) * XUC * UC(I)	00001690
	+ PINC * PIUP * (CCLH(I) * CLR + DCLH(I) * DLR)	00001700
	+ PIUP * (CCMP * ADCN * CDR + H) * PINC * FLOAT(M)	00001710
	* YOH * K * FLOAT(QPA(I)) * ((1.-RIP(I))	00001720
	* FLOAT(MCP(I)) + (1.-DS)) / MTBI(I)	00001730
4	IDFR(I) = I	00001740
5	CONTINUE	00001750
C	LD = LDPR	00001760
		00001770
		00001780
		00001790
		00001800
		00001810
		00001820
		00001830
		00001840
		00001850
		00001860
		00001870
		00001880
		00001890

```

      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)
      IF (LDFR.GT.0.OR.LDERV.GT.0) CALL TDSORT(TDFR,IDFR,LD,N)
C
      RETURN
      END
      SUBROUTINE DK
C
C*****
C*          SUBROUTINE DK
C* THIS SUBROUTINE CALCULATES THE DERIVATIVE OF LIFE CYCLE COST WITH
C* RESPECT TO K (THE RELIABILITY FACTOR WHICH CONVERTS PREDICTED
C* FAILURE RATES TO OPERATIONAL FAILURE RATES).
C*****
C          COMMON /TDK/ TDK
C
C          COMMON /TDFR/ IDFR(200),TDFR(200)
C
C          COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200),
+             MTBI(200),N,PA(200),QPA(200),QR(200),RM(200),
+             UC(200)
C          INTEGER QPA,QR
C          REAL MTBI
C
C          TDK = 0.
C          DO 5 I=1,N
C             IF (QPA(I).EQ.0) GO TO 5
C             TDK = TDK + TDFR(I)
C          5 CONTINUE
C
C          RETURN
C          END
C          SUBROUTINE DUC
C
C*****
C*          SUBROUTINE DUC
C* THIS SUBROUTINE CALCULATES THE DERIVATIVE OF LIFE CYCLE COST WITH
C* RESPECT TO UC (UNIT COST).
C* AFTER ALL DERIVATIVES HAVE BEEN CALCULATED, THE SUBROUTINE CALLS
C* SUBROUTINE TDSORT TO 'BUBBLE UP' TO THE TOP OF ARRAY IDUC THE
C* INDEX NUMBERS OF THE LDUC LARGEST DERIVATIVES.
C*****
C          COMMON /TDUC/ IDUC(200),TDUC(200)
C
C          COMMON /CNTL/ EXIT,ITER,MAXPMT,PRNT,REDO,REREAD
C          INTEGER EXIT,PRNT,REDO,REREAD
C
C          COMMON /RD1/ ADCM,ADPM,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,CONF,CPHI,
+             CPHP,CPWT,CRCT,DAA,DLR,DRCT,DS,
+             PAC(10),FCS,PSD,H,INC,K,N,MCRS,MXHRS,
+             MCP,NDP,NSP,OST,PIUP,PNE,PPRS,PRCDX,REFURB,RMC,
+             SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC,
+             TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR
C          INTEGER CADRE
C          REAL INC,K,MCRS,MXHRS
C
C          COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200),
+             MTBI(200),N,PA(200),QPA(200),QR(200),RM(200),
+             UC(200)
C          INTEGER QPA,QR
C          REAL MTBI

```



```

COMMON /SENS/ FINC, LDCMH, LDCRTS, LDDMH, LDDRTS, LDERV, LDFPR, LDPR,
* LDHCI, LDRH, LDSRTS, LDWOR, LDUC
C
COMMON /AUX1/ CCOND(200), COND(200), CRTS(200), DCOND(200),
* DRTS(200), SNRTS(200), SRTS(200), WR(200)
C
COMMON /AUX2/ YPR(200), WPR(200)
C
COMMON /AUX4/ CAS(200), DAS(200), SAS(200)
C
COMMON /AUX5/ DSTK(200), ECHT, MCF(200), STK(200), STK1(200), Z
INTEGER STK, STK1
C
DO 5 I=1, N
  IF (QPA(I).NE.0) GO TO 1
  TDUC(I) = 0.
  GO TO 4
1
  TDUC(I) = (FLOAT(N) * FLOAT(STK(I)) + AINT(CAS(I))
  * B * SORT(CAS(I)) + 0.5)
  * AINT(DAS(I)) + B * SORT(DAS(I)) + 0.5))
  * XUC * FINC * UC(I)
  * FLOAT(N-1) * FLOAT(QPA(I)) * XUC * FINC * UC(I)
  * PIUP * YPR(I) * (WR(I) + (1.-WR(I)) * RH(I))
  * XUC * FINC * UC(I)
4
  IDUC(I) = I
5
CONTINUE
C
LD = LDUC
IF (PRMT.NE.0) LD = MAX0(LD, LDERV)
CALL TDSORT(TDUC, IDUC, LD, N)
C
RETURN
END
SUBROUTINE DWOR
C*****
C* SUBROUTINE DWOR
C* THIS SUBROUTINE CALCULATES THE DERIVATIVE OF LIFE CYCLE COST WITH
C* RESPECT TO WOR (WEAR OUT RATE).
C* AFTER ALL DERIVATIVES HAVE BEEN CALCULATED, THE SUBROUTINE CALLS
C* SUBROUTINE TDSORT TO 'BUBBLE UP' TO THE TOP OF ARRAY IDWOR THE
C* INDEX NUMBERS OF THE LDWOR LARGEST DERIVATIVES.
C*****
COMMON /TDWOR/ IDWOR(200), TDWOR(200), WF(200)
C
COMMON /CNTL/ EXIT, ITER, MAXPMT, PRNT, REDO, REREAD
INTEGER EXIT, PRNT, REDO, REREAD
C
COMMON /RD4/ INO(200), LRNAME(200, 24), LRPART(200, 12), HCI(200),
* MTBI(200), N, PA(200), QPA(200), QR(200), RH(200),
* UC(200)
INTEGER QPA, QR
REAL MTBI
C
COMMON /RD5/ BCMH(200), CBCMH(200), CMH(200), CRTS1(200), CRTS2(200),
* DBCMh(200), DMH(200), DRTS1(200), DRTS2(200),
* DRTS3(200), FPR(200), IMH(200), INO15(200), N15,
* PAMH(200), RIP(200), RL(200), RMH(200), SMH(200), SRTS1(200),
* WOR1(200), WOR2(200), WOR3(200)
REAL IMH
INTEGER RL

```



```

COMMON /SENS/ FINC,LDCMH,LDCRTS,LDDMH,LDDRTS,LDERV,LDFPR,LDFR,
+ LDMCI,LDRH,LDSRTS,LDWOR,LDUC
C
COMMON /AUX1/ CCOND(200),COND(200),CRTS(200),DCOND(200),
+ DRTS(200),SNRTS(200),SRTS(200),WR(200)
C
COMMON /CHLCC/ CCF,CF,CHLCC,DCF,DF,SCF,SF
C
DO 6 I=1,N
  IF (QPA(I).NE.0) GO TO 1
  TDWOR(I) = 0.
  WF(I) = 0.
  GO TO 5
1  WF(I) = AMIN1(FINC,1.-WR(I))
  IF (WF(I).GT.0.00001) GO TO 2
  TDWOR(I) = 0.
  GO TO 5
2  CONTINUE
  SCF = 0.
  SF = (-1. * SRTS(I) * WF(I)) / (SRTS(I) + CRTS(I) + DRTS(I))
  CF = (-1. * CRTS(I) * WF(I)) / (SRTS(I) + CRTS(I) + DRTS(I))
  DF = (-1. * DRTS(I) * WF(I)) / (SRTS(I) + CRTS(I) + DRTS(I))
  IF (RL(I).EQ.3) GO TO 3
  CCF = WF(I)
  DCF = 0.
  GO TO 4
3  CCF = 0.
  DCF = WF(I)
4  CONTINUE
  CALL CHCALC(I)
  TDWOR(I) = CHLCC
5  CONTINUE
  IDWOR(I) = I
6  CONTINUE
C
LD = LDWOR
IF (PRNT.NE.0) LD = MAX0(LD,LDERV)
CALL TDSORT(TDWOR,IDWOR,LD,N)
C
RETURN
END
SUBROUTINE DFPR
C
C*****
C* SUBROUTINE DFPR
C* THIS SUBROUTINE CALCULATES THE DERIVATIVE OF LIFE CYCLE COST WITH
C* RESPECT TO FPR (FALSE PULL RATE).
C* AFTER ALL DERIVATIVES HAVE BEEN CALCULATED, THE SUBROUTINE CALLS
C* SUBROUTINE TDSORT TO 'BUBBLE UP' TO THE TOP OF ARRAY IDFPR THE
C* INDEX NUMBERS OF THE LDFPR LARGEST DERIVATIVES.
C*****
COMMON /TDFPR/ IDFPR(200),TDFPR(200)
C
COMMON /CNTL/ EXIT,ITER,MAXPMT,PRNT,REDO,REREAD
INTEGER EXIT,PRNT,REDO,REREAD
C
COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCHP,CDR,CDWH,CLR,CONF,CPNI,
+ CPMP,CPWT,CRCT,DAA,DLR,DRCT,DS,
+ FAC(10),FCS,FSD,H,INC,K,H,HCRS,MXHS,
+ MCP,NDP,NSP,OST,PIUP,PHE,PPRS,PRODX,REFURB,RNC,
+ SAA,SATBI,SOSI,SPRS,SRCT,SW,SWPAC,
+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR

```

```

      INTEGER CADRE
      REAL INC,K,MCRS,MKHSR
C
      COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),MCI(200),
      * MTBI(200),N,PA(200),QPA(200),QR(200),RH(200),
      * UC(200)
      INTEGER QPA,QR
      REAL MTBI
C
      COMMON /RD5/ BCMH(200),CBCMH(200),CHH(200),CRTS1(200),CRTS2(200),
      * DBCMH(200),DHH(200),DRTS1(200),DRTS2(200),
      * DRTS3(200),FPR(200),INH(200),INO15(200),N15,
      * PAMH(200),RIP(200),RL(200),RHH(200),SHH(200),SRTS1(200),
      * WOR1(200),WOR2(200),WOR3(200)
      REAL INH
      INTEGER RL
C
      COMMON /SENS/ PINC,LDCMH,LDCRTS,LDDHH,LDDRTS,LDERV,LDFPR,LDFR,
      * LDMCI,LDRH,LDSRTS,LDWOR,LDOC
C
      COMMON /AUX1/ CCOND(200),COND(200),CRTS(200),DCOND(200),
      * DRTS(200),SNRTS(200),SRTS(200),WR(200)
C
      COMMON /AUX2/ YFR(200),WFR(200)
C
      COMMON /AUX4/ CAS(200),DAS(200),SAS(200)
C
      COMMON /AUX5/ DSTK(200),ECMT,MCP(200),STK(200),STK1(200),Z
      INTEGER STK,STK1
C
      DO 5 I=1,N
      IF (QPA(I).NE.0) GO TO 1
      TDFPR(I) = 0.
      GO TO 4
1      TDFPR(I) = PINC * PIUP * YFR(I) * FPR(I) * (SNRTS(I) + COND(I))
      * CBCMH(I) * CLR
      * DSTK(I) * WFR(I) * PINC * FPR(I) * (SRTS(I) * SRCT
      * (1.- SRTS(I)) * SOSI) * XUC * UC(I)
      IF (CAS(I).GT..0001) X1 = (1.+ B / (2. * SQRT(CAS(I)))) * WFR(I)
      * PINC * FPR(I) * (1.- SRTS(I)) * CRCT * XUC * UC(I)
      IF (CAS(I).GT..0001) TDFPR(I) = TDFPR(I) + X1
4      IDFPR(I) = I
5      CONTINUE
C
      LD = LDFPR
      IF (PRNT.NE.0) LD = MAX0(LD,LDERV)
      CALL TDSORT(TDFPR,IDFPR,LD,N)
C
      RETURN
      END
      SUBROUTINE DCMH
C*****
C* SUBROUTINE DCMH
C* THIS SUBROUTINE CALCULATES THE DERIVATIVE OF LIFE CYCLE COST WITH
C* RESPECT TO CMH (AVERAGE MAN-HRS FOR CORRECTIVE MAINTENANCE AT CMH)
C* INCLUDING FAULT ISOLATION, REMOVAL, AND REPLACEMENT.
C* AFTER ALL DERIVATIVES HAVE BEEN CALCULATED, THE SUBROUTINE CALLS
C* SUBROUTINE TDSORT TO 'BUBBLE UP' TO THE TOP OF ARRAY IDCNH THE
C* INDEX NUMBERS OF THE LDCMH LARGEST DERIVATIVES.
C*****
      COMMON /TDCMH/ IDCNH(200),TDCMH(200)

```

C	COMMON /CNTL/ EXIT,ITER,MAXPMT,PRNT,REDO,BEREAD	00004410
	INTEGER EXIT,PRNT,REDO,BEREAD	00004420
C	COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,CONF,CPHI,	00004430
+	CPHP,CPWT,CRCT,DAA,DLR,DRCT,DS,	00004440
+	PAC(10),PCS,FSD,H,INC,K,H,HCRS,MXHS,	00004450
+	NCP,NDP,NSP,OST,PIUP,PNE,PPRS,PRODX,REFURB,RHC,	00004460
+	SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWPAC,	00004470
+	TC,TCCAD,TE,TB,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR	00004480
	INTEGER CADRE	00004490
	REAL INC,K,HCRS,MXHS	00004500
C	COMMON /RD4/ INO(200),LRNAME(200,24),LRPABT(200,12),NCI(200),	00004510
+	MTBI(200),N,PA(200),QPA(200),QR(200),RH(200),	00004520
+	UC(200)	00004530
	INTEGER QPA,QR	00004540
	REAL MTBI	00004550
C	COMMON /RD5/ BCMH(200),CBCMH(200),CMH(200),CRTS1(200),CRTS2(200),	00004560
+	DBCCH(200),DMH(200),DRTS1(200),DRTS2(200),	00004570
+	DRTS3(200),FPR(200),IMH(200),INO15(200),N15,	00004580
+	PAMH(200),RIP(200),RL(200),RMH(200),SMH(200),SRTS1(200),	00004590
+	WOR1(200),WOR2(200),WOR3(200)	00004600
	REAL IMH	00004610
	INTEGER RL	00004620
C	COMMON /SENS/ PINC,LDCMH,LDCRTS,LDDMH,LDDRTS,LDERV,LDFPR,LDFR,	00004630
+	LDHCI,LDRM,LDSRTS,LDWOR,LDUC	00004640
C	COMMON /AUX1/ CCOND(200),COND(200),CRTS(200),DCOND(200),	00004650
+	DRTS(200),SNRTS(200),SRTS(200),WR(200)	00004660
C	COMMON /AUX2/ YFR(200),WPR(200)	00004670
C	DO 5 I=1,N	00004680
	IF (QPA(I).NE.0) GO TO 1	00004690
	TDCMH(I) = 0.	00004700
	GO TO 4	00004710
1	TDCMH(I) = PIUP * YFR(I) * CRTS(I) * PINC * CMH(I) * CLR	00004720
4	IDCMH(I) = I	00004730
5	CONTINUE	00004740
C	LD = LDCMH	00004750
	IF (PRNT.NE.0) LD = MAXC(LD,LDERV)	00004760
	CALL TDSORT(TDCMH,IDCMH,LD,N)	00004770
C	RETURN	00004780
	END	00004790
	SUBROUTINE DDMH	00004800
C	*****	00004810
C*	SUBROUTINE DDMH	00004820
C*	THIS SUBROUTINE CALCULATES THE DERIVATIVE OF LIFE CYCLE COST WITH	00004830
C*	RESPECT TO DMH (AVERAGE MAN-HRS FOR CORRECTIVE MAINT. AT DEPOT).	00004840
C*	INCLUDING FAULT ISOLATION, REMOVAL, AND REPLACEMENT.	00004850
C*	AFTER ALL DERIVATIVES HAVE BEEN CALCULATED, THE SUBROUTINE CALLS	00004860
C*	SUBROUTINE TDSORT TO 'BUBBLE UP' TO THE TOP OF ARRAY IDDMH THE	00004870
C*	INDEX NUMBERS OF THE LDDMH LARGEST DERIVATIVES.	00004880
C*	*****	00004890
C	COMMON /TDDMH/ IDDMH(200),TDDMH(200)	00004900
C		00004910
C		00004920
C		00004930
C		00004940
C		00004950
C		00004960
C		00004970
C		00004980
C		00004990
C		00005000
C		00005010
C		00005020
C		00005030


```

COMMON /CNTL/ EXIT,ITER,HAIPT,PRNT,REDO,REREAD
INTEGER EXIT,PRNT,REDO,REREAD
00005040
00005050
00005060
C
COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCNP,CDR,CDWH,CLR,CONF,CPHI,
+ CPMP,CPNT,CRCT,DAA,DLR,DRCT,DS,
+ PAC(10),PCS,PSD,H,INC,K,H,HCRS,HXHS,
+ MCP,NDP,NSP,OST,PIUP,PNE,PPRS,PRODI,REFURB,RNC,
+ SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWPAC,
+ TC,TCCAD,TE,TB,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR
00005070
00005080
00005090
00005100
00005110
00005120
INTEGER CADRE
REAL INC,K,HCRS,HXHS
00005130
00005140
C
COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200),
+ MTBI(200),N,PA(200),QPA(200),QB(200),RM(200),
+ UC(200)
00005150
00005160
00005170
00005180
00005190
00005200
00005210
C
COMMON /RD5/ BCMH(200),CBCMH(200),CHH(200),CRTS1(200),CRTS2(200),
+ DBCMH(200),DHH(200),DRTS1(200),DRTS2(200),
+ DRTS3(200),FPR(200),IMH(200),INO15(200),W15,
+ PAHH(200),RIP(200),RL(200),RHH(200),SHH(200),SRTS1(200),
+ WOR1(200),WOR2(200),WOR3(200)
00005220
00005230
00005240
00005250
00005260
00005270
00005280
00005290
C
COMMON /SENS/ FINC,LDCMH,LDCRTS,LDDHH,LDDRTS,LDERV,LDPPR,LDPR,
+ LDMCI,LDRN,LDSRTS,LDWOR,LDUC
00005300
00005310
00005320
C
COMMON /AUX1/ CCOND(200),COND(200),CRTS(200),DCOND(200),
+ DRTS(200),SRTS(200),SRTS(200),WR(200)
00005330
00005340
00005350
00005360
C
COMMON /AUX2/ YPR(200),WPR(200)
00005370
00005380
00005390
00005400
00005410
1
4
5
TDDHH(I) = PIUP * YPR(I) * DRTS(I) * FINC * DHH(I) * DLR
IDDDHH(I) = I
CONTINUE
00005420
00005430
00005440
C
LD = LDDHH
IF (PRNT.NE.0) LD = MAX0(LD,LDERV)
CALL TDSORT(TDDHH,IDDDHH,LD,N)
00005450
00005460
00005470
00005480
00005490
00005500
00005510
00005520
00005530
C
SUBROUTINE DRN
00005540
C*
SUBROUTINE DRN
*00005550
C* THIS SUBROUTINE CALCULATES THE DERIVATIVE OF LIFE CYCLE COST WITH
*00005560
C* RESPECT TO RM (FRACTION OF UC(I) CONSUMED IN REPAIR OF LRU I).
*00005570
C* AFTER ALL DERIVATIVES HAVE BEEN CALCULATED, THE SUBROUTINE CALLS
*00005580
C* SUBROUTINE TDSORT TO 'BUBBLE UP' TO THE TOP OF ARRAY IDRN THE
*00005590
C* INDEX NUMBERS OF THE LDRN LARGEST DERIVATIVES.
*00005600
C*****
00005610
C
COMMON /TDRN/ IDRN(200),TDRN(200)
00005620
00005630
C
COMMON /CNTL/ EXIT,ITER,HAIPT,PRNT,REDO,REREAD
00005640
00005650
00005660
INTEGER EXIT,PRNT,REDO,REREAD

```



```

C      COMMON /RD1/ ADCM,ADPM,B,CAA,CADRE,CCHP,CDR,CDWH,CLR,CONF,CPHI,
+      CPHP,CPWT,CRCT,DAA,DLR,DRCT,DS,
+      PAC(10),FCS,FSD,H,INC,K,M,MCRS,MXHS,
+      NCP,NDP,NSP,OST,PIUP,PHE,PPRS,PRCDX,REFURB,RMC,
+      SA,SAA,SMTBI,SOSI,SPRS,SRCT,SW,SWPAC,
+      TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR
      INTEGER CADRE
      REAL INC,K,MCRS,MXHS
C
C      COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200),
+      MTBI(200),N,PA(200),QPA(200),QR(200),RM(200),
+      UC(200)
      INTEGER QPA,QR
      REAL MTBI
C
C      COMMON /SENS/ FINC,LDCMH,LDCRTS,LDDMH,LDDRTS,LDERV,LDFPR,LDFR,
+      LDMCI,LDRM,LDSRTS,LDWOR,LDUC
C
C      COMMON /AUX1/ CCOND(200),COND(200),CRTS(200),DCOND(200),
+      DRTS(200),SNRTS(200),SRTS(200),WR(200)
C
C      COMMON /AUX2/ YPR(200),WPR(200)
C
      DO 5 I=1,N
        IF (QPA(I).NE.0) GO TO 1
        TDRM(I) = 0.
        GO TO 4
1       TDRM(I) = PIUP * YPR(I) * (1.-WR(I)) * FINC * RM(I) * XUC *
+       UC(I)
4       IDRM(I) = I
5       CONTINUE
C
      LD = LDRM
      IF (PRNT.NE.0) LD = MAXO(LD,LDERV)
      CALL TDSORT(TDRM,IDRM,LD,N)
C
      RETURN
      END
      SUBROUTINE DSRTS
C*****
C*          SUBROUTINE DSRTS
C* THIS SUBROUTINE CALCULATES THE DERIVATIVE OF LIFE CYCLE COST WITH
C* RESPECT TO SRTS (FRACTION OF FAILURES REPAIRED AT SITE).
C* THE CALCULATION OF TDSRTS(I) IS DONE BY LETTING SUBROUTINE CHCALC
C* CALCULATE CHLCC = F(SF,CF,DP,SCF,CCF,DCF) FROM THE VALUES OF SF,CF,
C* DP,SCF,DCF,CCF THAT ARE SUPPLIED BY THIS SUBROUTINE AND THEN BY
C* SETTING TDSRTS(I) = CHLCC.
C* AFTER ALL DERIVATIVES HAVE BEEN CALCULATED, THE SUBROUTINE CALLS
C* SUBROUTINE TDSORT TO 'BUBBLE UP' TO THE TOP OF ARRAY IDSRTS THE
C* INDEX NUMBERS OF THE LDSRTS LARGEST DERIVATIVES.
C*****
C      COMMON /TDSRTS/ IDSRTS(200),SPSRTS(200),TDSRTS(200)
C
C      COMMON /CNTL/ EXIT,ITER,MAXPNT,PRNT,REDO,REREAD
      INTEGER EXIT,PRNT,REDO,REREAD
C
C      COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200),
+      MTBI(200),N,PA(200),QPA(200),QR(200),RM(200),
+      UC(200)
      INTEGER QPA,QR

```

	REAL MTBI	00006300
C	COMMON /SENS/ FINC, LDCMH, LDCRTS, LDDMH, LDDRTS, LDERV, LDFPR, LDFR,	00006310
	+ LDHCI, LDRH, LDSRTS, LDWOR, LDUC	00006320
		00006330
C	COMMON /AUX1/ CCOND(200), COND(200), CRTS(200), DCOND(200),	00006340
	+ DRTS(200), SNRTS(200), SRTS(200), WR(200)	00006350
		00006360
C	COMMON /CHLCC/ CCP, CF, CHLCC, DCF, DF, SCF, SF	00006370
		00006380
C	DO 6 I=1, N	00006390
	IF (QPA(I).NE.0) GO TO 1	00006400
	TDSRTS(I) = 0.	00006410
	SFSRTS(I) = 0.	00006420
	GO TO 5	00006430
1	SF = ANIN1(FINC, 1.- (CCOND(I) + DCOND(I) + SRTS(I)))	00006440
	SFSRTS(I) = SF	00006450
	IF (SF.GT.0.00001) GO TO 2	00006460
	TDSRTS(I) = 0.	00006470
	GO TO 5	00006480
2	CONTINUE	00006490
	DCF = 0.	00006500
	IF (COND(I).LT..99999) GO TO 3	00006510
	SCF = -1.	00006520
	CCF = 1. - SF	00006530
	CF = 0.	00006540
	DF = 0.	00006550
	GO TO 4	00006560
3	CCF = 0.	00006570
	SCF = 0.	00006580
	CF = (-1. * CRTS(I) * SF) / (CRTS(I) + DRTS(I))	00006590
	DF = (-1. * DRTS(I) * SF) / (CRTS(I) + DRTS(I))	00006600
4	CONTINUE	00006610
	CALL CHCALC(I)	00006620
	TDSRTS(I) = CHLCC	00006630
5	CONTINUE	00006640
	IDSRIS(I) = I	00006650
6	CONTINUE	00006660
		00006670
C	LD = LDSRTS	00006680
	IF (PRNT.NE.0) LD = MAX0(LD, LDERV)	00006690
	CALL TDSORT(TDSRTS, IDSRIS, LD, N)	00006700
		00006710
C	RETURN	00006720
	END	00006730
	SUBROUTINE DCRTS	00006740
		00006750
C		00006760
C*	*****	00006770
C*	SUBROUTINE DCRTS	00006780
C*	THIS SUBROUTINE CALCULATES THE DERIVATIVE OF LIFE CYCLE COST WITH	00006790
C*	RESPECT TO CRTS (FRACTION OF FAILURES REPAIRED AT THE CNP).	00006800
C*	THE CALCULATION OF TDCRTS(I) IS DONE BY LETTING SUBROUTINE CHCALC	00006810
C*	CALCULATE CHLCC = F(SF, CF, DF, SCF, CCP, DCF) FROM THE VALUES OF SF, CF,	00006820
C*	DF, SCF, DCF, CCP THAT ARE SUPPLIED BY THIS SUBROUTINE AND THEN BY	00006830
C*	SETTING TDCRTS(I) = CHLCC.	00006840
C*	AFTER ALL DERIVATIVES HAVE BEEN CALCULATED, THE SUBROUTINE CALLS	00006850
C*	SUBROUTINE TDSORT TO 'BUBBLE UP' TO THE TOP OF ARRAY IDCRTS THE	00006860
C*	INDEX NUMBERS OF THE LDCRTS LARGEST DERIVATIVES.	00006870
C*	*****	00006880
C		00006890
	COMMON /TDCRTS/ CFCRTS(200), IDCRTS(200), TDCRTS(200)	00006900
C		00006910
	COMMON /CNTL/ EXIT, ITER, MAXPNT, PRNT, REDO, REREAD	00006920

```

C      INTEGER EXIT,PRNT,REDO,REREAD                                00006930
C      COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200), 00006940
*      MTBI(200),N,PA(200),QPA(200),QR(200),RH(200),                00006950
*      HC(200)                                                        00006960
C      INTEGER QPA,QR                                                00006970
C      REAL MTBI                                                    00006980
C      COMMON /SENS/ FINC,LDCMH,LDCRTS,LDDMH,LDDRTS,LDERV,LDFPR,LDFR, 00006990
*      LDMCI,LDRH,LDSRTS,LDWOR,LDUC                                  00007000
C      COMMON /AUX1/ CCOND(200),COND(200),CRTS(200),DCOND(200),      00007010
*      DRTS(200),SNRTS(200),SRTS(200),WR(200)                      00007020
C      COMMON /CHLCC/ CCF,CF,CHLCC,DCF,DF,SCF,SF                    00007030
C      DO 6 I=1,N                                                    00007040
C          IF (QPA(I).NE.0) GO TO 1                                    00007050
C              TDCRTS(I) = 0.                                         00007060
C              CFCRTS(I) = 0.                                         00007070
C              GO TO 5                                                  00007080
1      CF = ANIN1(FINC,1.- (CCOND(I) + DCOND(I) + CRTS(I)))          00007090
C          CFCRTS(I) = CF                                              00007100
C          IF (CF.GT.0.00001) GO TO 2                                  00007110
C              TDCRTS(I) = 0.                                         00007120
C              GO TO 5                                                  00007130
2      CONTINUE                                                       00007140
C          DCF = 0.                                                    00007150
C          IF (COND(I).LT..99999) GO TO 3                              00007160
C          SCF = -1.                                                   00007170
C          CCF = 1. - CF                                               00007180
C          SF = 0.                                                     00007190
C          DF = 0.                                                     00007200
C          GO TO 4                                                      00007210
3      CCF = 0.                                                        00007220
C          SCF = 0.                                                    00007230
C          SF = (-1. * SRTS(I) * CF) / (SRTS(I) + DRTS(I))          00007240
C          DF = (-1. * DRTS(I) * CF) / (SRTS(I) + DRTS(I))          00007250
4      CONTINUE                                                       00007260
C          CALL CHCALC(I)                                              00007270
C          TDCRTS(I) = CHLCC                                           00007280
5      CONTINUE                                                       00007290
C          IDCRTS(I) = I                                              00007300
6      CONTINUE                                                       00007310
C      LD = LDCRTS                                                    00007320
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00007330
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00007340
C      RETURN                                                         00007350
C      END                                                            00007360
C      SUBROUTINE DDRTS                                              00007370
C      LD = LDCRTS                                                    00007380
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00007390
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00007400
C      RETURN                                                         00007410
C      END                                                            00007420
C      SUBROUTINE DDRTS                                              00007430
C      LD = LDCRTS                                                    00007440
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00007450
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00007460
C      RETURN                                                         00007470
C      END                                                            00007480
C      SUBROUTINE DDRTS                                              00007490
C      LD = LDCRTS                                                    00007500
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00007510
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00007520
C      RETURN                                                         00007530
C      END                                                            00007540
C      SUBROUTINE DDRTS                                              00007550
C      LD = LDCRTS                                                    00007560
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00007570
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00007580
C      RETURN                                                         00007590
C      END                                                            00007600
C      SUBROUTINE DDRTS                                              00007610
C      LD = LDCRTS                                                    00007620
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00007630
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00007640
C      RETURN                                                         00007650
C      END                                                            00007660
C      SUBROUTINE DDRTS                                              00007670
C      LD = LDCRTS                                                    00007680
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00007690
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00007700
C      RETURN                                                         00007710
C      END                                                            00007720
C      SUBROUTINE DDRTS                                              00007730
C      LD = LDCRTS                                                    00007740
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00007750
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00007760
C      RETURN                                                         00007770
C      END                                                            00007780
C      SUBROUTINE DDRTS                                              00007790
C      LD = LDCRTS                                                    00007800
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00007810
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00007820
C      RETURN                                                         00007830
C      END                                                            00007840
C      SUBROUTINE DDRTS                                              00007850
C      LD = LDCRTS                                                    00007860
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00007870
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00007880
C      RETURN                                                         00007890
C      END                                                            00007900
C      SUBROUTINE DDRTS                                              00007910
C      LD = LDCRTS                                                    00007920
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00007930
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00007940
C      RETURN                                                         00007950
C      END                                                            00007960
C      SUBROUTINE DDRTS                                              00007970
C      LD = LDCRTS                                                    00007980
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00007990
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008000
C      RETURN                                                         00008010
C      END                                                            00008020
C      SUBROUTINE DDRTS                                              00008030
C      LD = LDCRTS                                                    00008040
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008050
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008060
C      RETURN                                                         00008070
C      END                                                            00008080
C      SUBROUTINE DDRTS                                              00008090
C      LD = LDCRTS                                                    00008100
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008110
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008120
C      RETURN                                                         00008130
C      END                                                            00008140
C      SUBROUTINE DDRTS                                              00008150
C      LD = LDCRTS                                                    00008160
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008170
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008180
C      RETURN                                                         00008190
C      END                                                            00008200
C      SUBROUTINE DDRTS                                              00008210
C      LD = LDCRTS                                                    00008220
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008230
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008240
C      RETURN                                                         00008250
C      END                                                            00008260
C      SUBROUTINE DDRTS                                              00008270
C      LD = LDCRTS                                                    00008280
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008290
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008300
C      RETURN                                                         00008310
C      END                                                            00008320
C      SUBROUTINE DDRTS                                              00008330
C      LD = LDCRTS                                                    00008340
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008350
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008360
C      RETURN                                                         00008370
C      END                                                            00008380
C      SUBROUTINE DDRTS                                              00008390
C      LD = LDCRTS                                                    00008400
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008410
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008420
C      RETURN                                                         00008430
C      END                                                            00008440
C      SUBROUTINE DDRTS                                              00008450
C      LD = LDCRTS                                                    00008460
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008470
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008480
C      RETURN                                                         00008490
C      END                                                            00008500
C      SUBROUTINE DDRTS                                              00008510
C      LD = LDCRTS                                                    00008520
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008530
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008540
C      RETURN                                                         00008550
C      END                                                            00008560
C      SUBROUTINE DDRTS                                              00008570
C      LD = LDCRTS                                                    00008580
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008590
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008600
C      RETURN                                                         00008610
C      END                                                            00008620
C      SUBROUTINE DDRTS                                              00008630
C      LD = LDCRTS                                                    00008640
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008650
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008660
C      RETURN                                                         00008670
C      END                                                            00008680
C      SUBROUTINE DDRTS                                              00008690
C      LD = LDCRTS                                                    00008700
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008710
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008720
C      RETURN                                                         00008730
C      END                                                            00008740
C      SUBROUTINE DDRTS                                              00008750
C      LD = LDCRTS                                                    00008760
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008770
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008780
C      RETURN                                                         00008790
C      END                                                            00008800
C      SUBROUTINE DDRTS                                              00008810
C      LD = LDCRTS                                                    00008820
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008830
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008840
C      RETURN                                                         00008850
C      END                                                            00008860
C      SUBROUTINE DDRTS                                              00008870
C      LD = LDCRTS                                                    00008880
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008890
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008900
C      RETURN                                                         00008910
C      END                                                            00008920
C      SUBROUTINE DDRTS                                              00008930
C      LD = LDCRTS                                                    00008940
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00008950
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00008960
C      RETURN                                                         00008970
C      END                                                            00008980
C      SUBROUTINE DDRTS                                              00008990
C      LD = LDCRTS                                                    00009000
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009010
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009020
C      RETURN                                                         00009030
C      END                                                            00009040
C      SUBROUTINE DDRTS                                              00009050
C      LD = LDCRTS                                                    00009060
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009070
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009080
C      RETURN                                                         00009090
C      END                                                            00009100
C      SUBROUTINE DDRTS                                              00009110
C      LD = LDCRTS                                                    00009120
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009130
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009140
C      RETURN                                                         00009150
C      END                                                            00009160
C      SUBROUTINE DDRTS                                              00009170
C      LD = LDCRTS                                                    00009180
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009190
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009200
C      RETURN                                                         00009210
C      END                                                            00009220
C      SUBROUTINE DDRTS                                              00009230
C      LD = LDCRTS                                                    00009240
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009250
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009260
C      RETURN                                                         00009270
C      END                                                            00009280
C      SUBROUTINE DDRTS                                              00009290
C      LD = LDCRTS                                                    00009300
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009310
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009320
C      RETURN                                                         00009330
C      END                                                            00009340
C      SUBROUTINE DDRTS                                              00009350
C      LD = LDCRTS                                                    00009360
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009370
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009380
C      RETURN                                                         00009390
C      END                                                            00009400
C      SUBROUTINE DDRTS                                              00009410
C      LD = LDCRTS                                                    00009420
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009430
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009440
C      RETURN                                                         00009450
C      END                                                            00009460
C      SUBROUTINE DDRTS                                              00009470
C      LD = LDCRTS                                                    00009480
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009490
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009500
C      RETURN                                                         00009510
C      END                                                            00009520
C      SUBROUTINE DDRTS                                              00009530
C      LD = LDCRTS                                                    00009540
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009550
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009560
C      RETURN                                                         00009570
C      END                                                            00009580
C      SUBROUTINE DDRTS                                              00009590
C      LD = LDCRTS                                                    00009600
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009610
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009620
C      RETURN                                                         00009630
C      END                                                            00009640
C      SUBROUTINE DDRTS                                              00009650
C      LD = LDCRTS                                                    00009660
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009670
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009680
C      RETURN                                                         00009690
C      END                                                            00009700
C      SUBROUTINE DDRTS                                              00009710
C      LD = LDCRTS                                                    00009720
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009730
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009740
C      RETURN                                                         00009750
C      END                                                            00009760
C      SUBROUTINE DDRTS                                              00009770
C      LD = LDCRTS                                                    00009780
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009790
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009800
C      RETURN                                                         00009810
C      END                                                            00009820
C      SUBROUTINE DDRTS                                              00009830
C      LD = LDCRTS                                                    00009840
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009850
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009860
C      RETURN                                                         00009870
C      END                                                            00009880
C      SUBROUTINE DDRTS                                              00009890
C      LD = LDCRTS                                                    00009900
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009910
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009920
C      RETURN                                                         00009930
C      END                                                            00009940
C      SUBROUTINE DDRTS                                              00009950
C      LD = LDCRTS                                                    00009960
C      IF (PRNT.NE.0) LD = MAXC(LD,LDERV)                             00009970
C      CALL TDSORT(TDCRTS,IDCRTS,LD,N)                                00009980
C      RETURN                                                         00009990
C      END                                                            00010000

```



```

C* INDEX NUMBERS OF THE LDDRTS LARGEST DERIVATIVES. *00007560
C*****00007570
C      COMMON /TDDRTS/ DFDRTS(200),IDDRTS(200),TDDRTS(200) 00007580
C      COMMON /CNTL/ EXIT,ITER,MAXPNT,PRNT,REDO,REREAD 00007590
C      INTEGER EXIT,PRNT,REDO,REREAD 00007600
C      COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),MCI(200), 00007610
C      + MTBI(200),N,PA(200),QPA(200),QR(200),RH(200), 00007620
C      + UC(200) 00007630
C      INTEGER QPA,QR 00007640
C      REAL MTBI 00007650
C      COMMON /SENS/ FINC,LDCNH,LDCRTS,LDDMH,LDDRTS,LDERV,LDFPR,LDFR, 00007660
C      + LDNCI,LDRH,LDSRTS,LDWOR,LDUC 00007670
C      COMMON /AUX1/ CCOND(200),COND(200),CRTS(200),DCOND(200), 00007680
C      + DRTS(200),SNRTS(200),SRTS(200),WR(200) 00007690
C      COMMON /CHLCC/ CCF,CF,CHLCC,DCF,DF,SCF,SP 00007700
C      DO 6 I=1,N 00007710
C      IF (QPA(I).NE.0) GO TO 1 00007720
C      TDDRTS(I) = 0. 00007730
C      DFDRTS(I) = 0. 00007740
C      GO TO 5 00007750
1    DF = AMIN1(FINC,1.- (CCOND(I) + DCOND(I) + DRTS(I))) 00007760
C      DFDRTS(I) = DF 00007770
C      IF (DF.GT.0.00001) GO TO 2 00007780
C      TDDRTS(I) = 0. 00007790
C      GO TO 5 00007800
2    CONTINUE 00007810
C      CCF = 0. 00007820
C      IF (COND(I).LT..99999) GO TO 3 00007830
C      SCF = -1. 00007840
C      DCF = 1. - DF 00007850
C      SF = 0. 00007860
C      CF = 0. 00007870
C      GO TO 4 00007880
3    DCF = 0. 00007890
C      SCF = 0. 00007900
C      SF = (-1. * SRTS(I) * DF) / (SRTS(I) + CRTS(I)) 00007910
C      CF = (-1. * CRTS(I) * DF) / (SRTS(I) + CRTS(I)) 00007920
4    CONTINUE 00007930
C      CALL CHCALC(I) 00007940
C      TDDRTS(I) = CHLCC 00007950
5    CONTINUE 00007960
C      IDDRTS(I) = I 00007970
6    CONTINUE 00007980
C      LD = LDDRTS 00007990
C      IF (PRNT.NE.0) LD = MAX0(LD,LDERV) 00008000
C      CALL TDSORT(TDDRTS,IDDRTS,LD,N) 00008010
C      RETURN 00008020
C      END 00008030
C      SUBROUTINE DMCI 00008040
C      *****00008050
C      SUBROUTINE DMCI *00008150
C* THIS SUBROUTINE CALCULATES THE DERIVATIVE OF LIFE CYCLE COST WITH *00008160
C* RESPECT TO MCI (MISSION CRITICAL INDICATOR). *00008170
C* *00008180

```



```

C* AFTER ALL DERIVATIVES HAVE BEEN CALCULATED, THE SUBROUTINE CALLS *00008190
C* SUBROUTINE TDSORT TO 'BUBBLE UP' TO THE TOP OF ARRAY IDNCI THE *00008200
C* INDEX NUMBERS OF THE LDNCI LARGEST DERIVATIVES. *00008210
C*****00008220
C      COMMON /TDMCI/ IDNCI(200),MCIC(200),TDMCI(200) 00008230
C      COMMON /CNTRL/ EXIT,ITFR,MAXPMT,PRNT,REDO,REREAD 00008240
C      INTEGER EXIT,PRNT,REDO,REREAD 00008250
C      COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,CONF,CPHI, 00008260
C      + CPMP,CPMT,CRCT,DAA,DLR,DRCT,DS, 00008270
C      + PAC(10),PCS,FSD,H,INC,K,H,MCRS,MXHRS, 00008280
C      + MCP,NDP,NSP,OST,PIUP,PME,PPRS,PRODX,REFURB,RMC, 00008290
C      + SA,SA,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC, 00008300
C      + TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR 00008310
C      INTEGER CADRE 00008320
C      REAL INC,K,MCRS,MXHRS 00008330
C      COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200), 00008340
C      + MTBI(200),N,PA(200),QPA(200),QR(200),RM(200), 00008350
C      + UC(200) 00008360
C      INTEGER QPA,QR 00008370
C      REAL MTBI 00008380
C      COMMON /SENS/ PINC,LDCMH,LDCRTS,LDDNH,LDDRTS,LDERV,LDFPR,LDFR, 00008390
C      + LDNCI,LDRM,LDSRTS,LDWOR,LDUC 00008400
C      COMMON /AUX2/ YFR(200),WFR(200) 00008410
C      COMMON /AUX5/ DSTK(200),ECMT,MCF(200),STK(200),STK1(200),Z 00008420
C      INTEGER STK,STK1 00008430
C      DO 5 I=1,N 00008440
C      MCIC(I) = 0 00008450
C      IF (QPA(I).NE.0) GO TO 1 00008460
C      TDMCI(I) = 0. 00008470
C      GO TO 4 00008480
C      1 L1 = STK1(I) + QR(I) 00008490
C      IF (HCI(I).EQ.1) MCIC(I) = 1 00008500
C      IF (HCI(I).EQ.2) MCIC(I) = -1 00008510
C      IF (L1.EQ.0.AND.MCIC(I).NE.0) GO TO 2 00008520
C      TDMCI(I) = 0. 00008530
C      GO TO 4 00008540
C      2 CONTINUE 00008550
C      IF (HCI(I).EQ.2) MCC = 1. 00008560
C      IF (HCI(I).EQ.1) MCC = -1. 00008570
C      TDMCI(I) = PIUP * YFR(I) * FLOAT(MCC) * (CCMP * ADCH * CDR + H) 00008580
C      - FLOAT(MCC) * FLOAT(N) * XUC * UC(I) 00008590
C      4 IDNCI(I) = I 00008600
C      5 CONTINUE 00008610
C      LD = LDNCI 00008620
C      IF (PRNT.NE.0) LD = MAX0(LD,LDERV) 00008630
C      CALL TDSORT(TDMCI,IDNCI,LD,N) 00008640
C      RETURN 00008650
C      END 00008660
C      SUBROUTINE CHCALC(I) 00008670
C*****00008680
C      SUBROUTINE CHCALC *00008690
C* BECAUSE OF GREAT SIMILARITIES IN THE EQUATIONS FOR THE DERIVATIVES *00008700

```

```

C* OF TDSRTS, TDCRTS, TDDRTS, AND TDWOR, THE PROGRAM USES SUBROUTINE *00008810
C* CHCALC TO CALCULATE THE FOUR DERIVATIVES. THE FOUR CALLING *00008820
C* ROUTINES (DSRTS, DCRTS, DDRTS, DWOR) SET VARIABLES SP, CF, DF, DCF, SCF, *00008830
C* AND CCF BEFORE CALLING CHCALC. CHCALC USED THESE SIX VARIABLES AS *00008840
C* PARAMETERS TO A SINGLE FUNCTION TO SET VARIABLE CHLCC TO THE VALUE *00008850
C* OF THE DERIVATIVE. HENCE, CHLCC = F(CF, SP, DF, CCF, DCF, SCF). *00008860
C*****00008870
C      COMMON /CHLCC/ CCF, CF, CHLCC, DCF, DF, SCF, SF      00008880
C      COMMON /RD1/ ADCN, ADPH, B, CAA, CADRE, CCHP, CDR, CDWH, CLR, COMP, CPNI,      00008890
C      + CPNP, CPWT, CRCT, DAA, DLR, DRCT, DS,      00008900
C      + PAC(10), PCS, PSD, H, INC, K, M, HCRS, MXHRS,      00008920
C      + NCP, NDP, NSP, OST, PIUP, PME, PPRS, PRODX, RPFURE, RMC,      00008930
C      + SA, SAA, SHTBI, SOSI, SPRS, SRCT, SW, SWFAC,      00008940
C      + TC, TCCAD, TE, TR, TW, TWCAD, VAL, XUC, YOH, YSCAD, YSLR      00008950
C      INTEGER CADRE      00008960
C      REAL INC, K, HCRS, MXHRS      00008970
C      COMMON /RD4/ INO(200), LRNAME(200, 24), LRPART(200, 12), MCI(200),      00008980
C      + MTBI(200), N, PA(200), QPA(200), QR(200), RH(200),      00008990
C      + UC(200)      00009000
C      INTEGER QPA, QR      00009010
C      REAL MTBI      00009020
C      COMMON /RD5/ BCHH(200), CBCHH(200), CHH(200), CRTS1(200), CRTS2(200),      00009030
C      + DBCHH(200), DMH(200), DRTS1(200), DRTS2(200),      00009040
C      + DRTS3(200), FPR(200), IMH(200), INO15(200), N15,      00009050
C      + PANH(200), RIP(200), RL(200), RMH(200), SHH(200), SRTS1(200),      00009060
C      + WOR1(200), WOR2(200), WOR3(200)      00009070
C      REAL IMH      00009080
C      INTEGER RL      00009090
C      COMMON /AUX2/ YPR(200), WPR(200)      00009100
C      COMMON /AUX4/ CAS(200), DAS(200), SAS(200)      00009110
C      COMMON /AUX5/ DSTK(200), ECRT, MCF(200), STK(200), STK1(200), Z      00009120
C      INTEGER STK, STK1      00009130
C      X1 = PIUP * YPR(I) * ((1. + FPR(I)) * (-1. * (SF + SCF)) * CBCHH(I)      00009140
C      + CF * CHH(I) + FPR(I) * SCF * CBCHH(I)) * CLR      00009150
C      + ((DCF + DF) * DBCHH(I) + DF * DMH(I)) * DLR      00009160
C      + DSTK(I) * WPR(I) * (1. + FPR(I)) * SF * (SRCT - SOSI)      00009170
C      + XUC * UC(I)      00009180
C      IF (CAS(I) .LE. 0.0001) X2 = 0.      00009190
C      IF (CAS(I) .GT. 0.0001) X2 = (1. + B / (2. * SQRT(CAS(I)))) * WPR(I)      00009200
C      + (FPR(I) * (-1.) * SF * CRCT + CF * CRCT      00009210
C      + (CCF + DF + DCF) * (OST + SOSI / 2.) + SCF * OST)      00009220
C      + XUC * UC(I)      00009230
C      IF (DAS(I) .LE. 0.0001) X3 = 0.      00009240
C      IF (DAS(I) .GT. 0.0001) X3 = (1. + B / (2. * SQRT(DAS(I)))) * WPR(I)      00009250
C      + DF * DRCT * XUC * UC(I)      00009260
C      X4 = PIUP * YPR(I) * (SCF + CCF + DCF) * (1. - RH(I)) * XUC * UC(I)      00009270
C      CHLCC = X1 + X2 + X3 + X4      00009280
C      RETURN      00009290
C      END      00009300
C      SUBROUTINE TDSORT(TD, ID, LD, N)      00009310
C*****00009320
C*      SUBROUTINE TDSORT      00009330
C*      THIS SUBROUTINE 'BUBBLES UP' TO THE FIRST -LD- POSITIONS IN ARRAY *00009340

```

C*	-ID- THE LRU INDEX NUMBERS CORRESPONDING TO THE -LD- HIGHEST	*00009440
C*	VALUES OF ARRAY -TD-. HENCE, AT THE END OF TDSORT, ID(1) HOLDS	*00009450
C*	THE INDEX NUMBER OF THE LRU WITH THE LARGEST DERIVATIVE, ID(2) THE	*00009460
C*	INDEX NUMBER OF THE SECOND LARGEST DERIVATIVE, ETC.	*00009470
C*****		*00009480
C		00009490
	DIMENSION TD(200),ID(200)	00009500
C		00009510
	DO 7 L=1,LD	00009520
	MA = N - L	00009530
	DO 6 M=1,MA	00009540
	MB = N - M	00009550
	IF (ABS(TD(ID(MB + 1))).LT.ABS(TD(ID(MB)))) GO TO 6	00009560
	IDD = ID(MB + 1)	00009570
	ID(MB + 1) = ID(MB)	00009580
	ID(MB) = IDD	00009590
6	CONTINUE	00009600
7	CONTINUE	00009610
C		00009620
	RETURN	00009630
	END	00009640


```

SUBROUTINE OTAB1                                00000010
C                                                    00000020
C*****00000030
C* SUBROUTINE OTAB1 PRINTS OUTPUT TABLE 1. THIS *00000040
C* TABLE CONSISTS OF SYSTEM TOTALS IN MILLIONS OF DOLLARS. ERROR *00000050
C* MESSAGES ARE PRINTED BENEATH THE TABLE IN THE EVENT THAT CERTAIN *00000060
C* DESIGNATED DATA CONSTRAINTS ARE NOT MET. *00000070
C*****00000080
C                                                    00000090
COMMON /CNTL/ EXIT,ITER,MAXPMT,PRMT,REDO,REREAD
INTEGER EXIT,PRMT,REDO,REREAD                    00000100
C                                                    00000110
C                                                    00000120
COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCHP,CDR,CDWH,CLR,CONF,CPHI,
+ CPHP,CPWT,CRCT,DAA,DLR,DRCT,DS,
+ FAC(10),FCS,FSD,H,INC,K,M,MCRS,MXHS,
+ NCP,NDP,NSP,OST,PIUP,PHE,PPRS,PRODX,REFURB,RNC,
+ SA,SAI,SMTBI,SOSI,SPRS,SRCT,SW,SWFAC,
+ TC,TCCAD,TE,TR,TW,TWCAD,VAL,XUC,YOH,YSCAD,YSLR
INTEGER CADRE
RFAL INC,K,MCRS,MXHS                            00000190
C                                                    00000200
C                                                    00000210
COMMON /NCOS/ CLH,PHFA(10),SMHH,SPMH,STHH
C                                                    00000220
C                                                    00000230
COMMON /AUX3/ CCLH(200),DCLH(200),DLH
C                                                    00000240
C                                                    00000250
COMMON /C1/ C1
C                                                    00000260
C                                                    00000270
COMMON /C2/ C2,C2C,C2D,C2L(200),C2S
C                                                    00000280
C                                                    00000290
COMMON /C3/ C3,C3C,C3D,C3L(200),C3S
C                                                    00000300
C                                                    00000310
COMMON /C4/ C4,C4I,C4L(200),C4R
C                                                    00000320
C                                                    00000330
COMMON /C5/ C5
C                                                    00000340
C                                                    00000350
COMMON /C6/ C6,C6C,C6D,C6I,C6R,C6S
C                                                    00000360
C                                                    00000370
COMMON /C7/ C7
C                                                    00000380
C                                                    00000390
COMMON /C8/ C8,C8C,C8D,C8I,C8R,C8S
C                                                    00000400
C                                                    00000410
COMMON /C9/ C9,C9C,C9D,C9I,C9L(200),C9R,C9S
C                                                    00000420
C                                                    00000430
COMMON /C10/ C10,C10HW(10),C10I,C10R,C10SW(10)
C                                                    00000440
C                                                    00000450
1 FORMAT (1H1,24X,3CHOUTPUT TABLE 1: SYSTEM TOTALS/31X,
+ 15H($ IN MILLIONS) /)
2 FORMAT (1H0)
3 FORMAT (4X,12HPROGRAM COST,6X,11HDEVELOPMENT,2X,
+ 10HPRODUCTION,3X,7HSUPPORT,4X,5HTOTAL,2X,6H % HW //
+ 24H PRIME MISSION EQUIPMENT/
+ 11H HARDWARE,11X,9H 0.0 ,F12.2,7X,4H0.0 ,F11.2,F7.1/
+ 11H SOFTWARE,16X,4H0.0 ,F12.2,7X,4H0.0 ,F11.2,F7.1/
+ 16H REFURBISHMENT,11X,4H0.0 ,F12.2,7X,4H0.0 ,F11.2,F7.1/
+ 18H DESIGN VALIDATION,4X,F9.2,8X,4H0.0 ,7X,4H0.0 ,F11.2,F7.1/
+ 23H FULL SCALE DEVELOPMENT,F8.2,8X,4H0.0 ,7X,4H0.0 ,F11.2,F7.1/
+ 17H ASSOCIATED COSTS,10X,4H0.0 ,F12.2,7X,4H0.0 ,F11.2,F7.1/
+ 9H SPARES ,18X,4H0.0 ,F12.2,F11.2,F7.1/
+ 12H MAINTENANCE,15X,4H0.0 ,8X,4H0.0 ,F11.2,F11.2,F7.1/
+ 12H CONSUMABLES,15X,4H0.0 ,8X,4H0.0 ,F11.2,F11.2,F7.1/
+ 18H SUPPORT EQUIPMENT,9X,4H0.0 ,F12.2,F11.2,F11.2,F7.1/
+ 11H FACILITIES,16X,4H0.0 ,F12.2,7X,4H0.0 ,F11.2,F7.1/
+ 9H TRAINING,18X,4H0.0 ,F12.2,F11.2,F11.2,F7.1/
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630

```



```

+ 21H INVENTORY MANAGEMENT,6X,4H0.0 ,F12.2,F11.2,F11.2,F7.1/ 00000640
+ 21H SOFTWARE MAINTENANCE,6X,4H0.0 ,F12.2,F11.2,F11.2,F7.1// 00000650
+ 4X,1HTOTAL COST:,4X,2F12.2,2F11.2/) 00000660
C 00000670
4 FORMAT (40H ***** ERROR. THE FOLLOWING CONSTRAINT , 00000680
+ 18HWAS NOT MET: *****) 00000690
C 00000700
5 FORMAT (40H ANNUAL ON-RADAR MAINT. MAN-HRS BY SITE , 00000710
+ 10HPERSONNEL:,F11.0,3H <=,F7.0) 00000720
C 00000730
6 FORMAT (41H ANNUAL OFF-RADAR MAINT. MAN-HRS BY SITE , 00000740
+ 10HPERSONNEL:,F10.0) 00000750
C 00000760
7 FORMAT (29X,6HTOTAL:,15X,F11.0,3H <=,F7.0/) 00000770
C 00000780
8 FORMAT (35H ANNUAL INTERMEDIATE (CMP) MAN-HRS:,15X,F11.0, 00000790
+ 3H <=,F7.1) 00000800
C 00000810
9 FORMAT (34H ANNUAL DEPOT MAINTENANCE MAN-HRS:,16X,F11.0, 00000820
+ 3H <=,F7.0) 00000830
C 00000840
C***** 00000850
C* COMPUTE ALL COSTS INVOLVING EQUATIONS AS WELL AS  $\Sigma$ (PME*XUC) COSTS.* 00000860
C***** 00000870
C 00000880
TEN1A = 100. 00000890
TEN1B = (SW / (PME * XUC)) * 100. 00000900
TEN1C = (REFURB / (PME * XUC)) * 100. 00000910
TEN1D = (VAL / (PME * XUC)) * 100. 00000920
TEN1E = (FSD / (PME * XUC)) * 100. 00000930
TEN1F = (PRODX / (PME * XUC)) * 100. 00000940
TEN2 = (C3+C4I) / 1000000. 00000950
TEN3 = (C3+C4) / 1000000. 00000960
TEN4 = ((C3+C4)/(PME*XUC)) * 100 00000970
TEN5 = (C2/(PME*XUC)) * 100 00000980
TEN6 = (C5/(PME*XUC)) * 100 00000990
TEN7 = (C6/(PME*XUC)) * 100 00001000
TEN8 = (C7/(PME*XUC)) * 100 00001010
TEN9 = (C8/(PME*XUC)) * 100 00001020
TEN10 = (C9/(PME*XUC)) * 100 00001030
TEN11 = (C10/(PME*XUC)) * 100 00001040
C 00001050
C***** 00001060
C* COMPUTE DEVELOPMENT, PRODUCTION, SUPPORT, AND SYSTEM TOTAL *00001070
C* COST IN MILLIONS OF DOLLARS. *00001080
C***** 00001090
C 00001100
TOT1 = (VAL + FSD) / 1000000. 00001110
TOT2 = (PME*XUC+SW+REFURB+PRODX+C3+C4I+C6I+C7+C8I+C9I+C10I) / 00001120
+ 1000000. 00001130
TOT3 = (C4R+C2+C5+C6R+C8R+C9R+C10R) / 1000000. 00001140
TOT4 = (C1+C3+C4+C2+C5+C6+C7+C8+C9+C10) / 1000000. 00001150
C 00001160
C***** 00001170
C* TRANSFORM ALL COSTS TO MILLIONS OF DOLLARS. *00001180
C***** 00001190
C 00001200
T1A1 = (PME * XUC) / 1000000. 00001210
T1B1 = SW / 1000000. 00001220
T1C1 = REFURB / 1000000. 00001230
T1D1 = VAL / 1000000. 00001240
T1E1 = FSD / 1000000. 00001250
T1F1 = PRODX / 1000000. 00001260

```

```

T4R = C4R / 1000000.
T2 = C2 / 1000000.
T5 = C5 / 1000000.
T6I = C6I / 1000000.
T6R = C6R / 1000000.
T6 = C6 / 1000000.
T7 = C7 / 1000000.
T8I = C8I / 1000000.
T8R = C8R / 1000000.
T8 = C8 / 1000000.
T9I = C9I / 1000000.
T9R = C9R / 1000000.
T9 = C9 / 1000000.
T10I = C10I / 1000000.
T10R = C10R / 1000000.
T10 = C10 / 1000000.

C
C*****
C* WRITE OUTPUT TABLE 1 AT THE TERMINAL IF PRNT = 1.
C*****
C
IF (PRNT.EQ.1) GO TO 111
WRITE (6,1)
WRITE (6,3) T1A1,T1A1,TEM1A,T1B1,T1B1,TEM1B,T1C1,T1C1,TEM1C,T1D1,
+ T1D1,TEM1D,T1E1,T1E1,TEM1E,T1F1,T1F1,TEM1F,
+ TEM2,T4R,TEM3,TEM4,T2,T2,TEM5,T5,
+ T5,TEM6,T6I,T6R,T6,TEM7,T7,T7,TEM8,T8I,T8R,T8,TEM9,T9I,T9R,
+ T9,TEM10,T10I,T10R,T10,TEM11,TOT1,TOT2,TOT3,TOT4
C
C*****
C* FOUR ERROR CHECKS ARE MADE HERE WITH MESSAGES PRINTED IN THE EVENT
C* THAT CONSTRAINTS ARE NOT MET. THE FIRST ERROR OCCURS WHEN THE
C* NUMBER OF SITE MAINTENANCE MAN-HOURS EXCEEDS THE MAXIMUM ON-RADAR
C* MAINTENANCE TIME.
C*****
C
IF (SMNH.GT.MXHRS) WRITE (6,4)
WRITE (6,5) SMNH,MXHRS
WRITE (6,6) SPNH
C
C*****
C* THE SECOND ERROR IS PRINTED WHEN THE TOTAL MAINTENANCE MAN-HOURS
C* BY SITE PERSONNEL EXCEEDS THE ANNUAL OFF-RADAR MAINTENANCE TIME.
C*****
C
TEM12 = FLOAT(NSP)*SAA
IF (STMH.GT.TEM12) WRITE (6,4)
WRITE (6,7) STMH,TEM12
C
C*****
C* THE THIRD ERROR IS PRINTED WHEN THE YEARLY MAINTENANCE HOURS BY
C* CNP PERSONNEL EXCEEDS THE TOTAL AMOUNT OF MAN-HOURS AVAILABLE AT
C* THE CNP.
C*****
C
TEM13 = FLOAT(NCP)*CAA
IF (CLH.GT.TEM13) WRITE (6,4)
IF (CLH.GT.TEM13) WRITE (6,8) CLH,TEM13
C
C*****
C* THE FOURTH ERROR IS PRINTED WHEN THE YEARLY CORRECTIVE MAINTENANCE
C* HOURS BY DEPOT PERSONNEL EXCEEDS THE TOTAL AMOUNT OF MAN-HOURS
C* AVAILABLE AT THE DEPOT.

```

```

C*****00001900
C                                00001910
    TEM14 = FLOAT(NDP)*DAA      00001920
    IF (DLH.GT.TEM14) WRITE (6,4) 00001930
    IF (DLH.GT.TEM14) WRITE (6,9) DLH,TEM14 00001940
C                                00001950
C*****00001960
C* WRITE OUTPUT TABLE 1 AT ON OFF-LINE PRINTER IF PRNT /= 0. *00001970
C*****00001980
C                                00001990
    111 IF (PRNT.EQ.0) GO TO 222 00002000
        WRITE (7,1)              00002010
        WRITE (7,2)              00002020
        WRITE (7,3) T1A1,T1A1,TEM1A,T1B1,T1B1,TEM1B,T1C1,T1C1,TEM1C,T1D1,
        * T1D1,TEM1D,T1E1,T1E1,TEM1E,T1F1,T1F1,TEM1F, 00002030
        * TEM2,T4R,TEM3,TEM4,T2,T2,TEM5,T5, 00002040
        * T5,TEM6,T6I,T6R,T6,TEM7,T7,T7,TEM8,T8I,T8R,T8,TEM9,T9I,T9R, 00002050
        * T9,TEM10,T10I,T10R,T10,TEM11,TOT1,TOT2,TOT3,TOT4 00002060
C                                00002070
C                                00002080
C*****00002090
C* FOUR ERROR CHECKS ARE MADE HERE WITH MESSAGES PRINTED IN THE EVENT *00002100
C* THAT CONSTRAINTS ARE NOT MET. THE FIRST ERROR OCCURS WHEN THE *00002110
C* NUMBER OF SITE MAINTENANCE MAN-HOURS EXCEEDS THE MAXIMUM ON-RADAR *00002120
C* MAINTENANCE TIME. *00002130
C*****00002140
C                                00002150
        IF (SMHH.GT.MXHRS) WRITE (7,4) 00002160
        WRITE (7,5) SMHH,MXHRS 00002170
        WRITE (7,6) SPMH 00002180
C                                00002190
C*****00002200
C* THE SECOND ERROR IS PRINTED WHEN THE TOTAL MAINTENANCE MAN-HOURS *00002210
C* BY SITE PERSONNEL EXCEEDS THE ANNUAL OFF-RADAR MAINTENANCE TIME. *00002220
C*****00002230
C                                00002240
        TEM12 = FLOAT(NSP)*SAA 00002250
        IF (SMHH.GT.TEM12) WRITE (7,4) 00002260
        WRITE (7,7) SMHH,TEM12 00002270
C                                00002280
C*****00002290
C* THE THIRD ERROR IS PRINTED WHEN THE YEARLY MAINTENANCE HOURS BY *00002300
C* CHF PERSONNEL EXCEEDS THE TOTAL AMOUNT OF MAN-HOURS AVAILABLE AT *00002310
C* THE DEPOT. *00002320
C*****00002330
C                                00002340
        TEM13 = FLOAT(NCP)*CAA 00002350
        IF (CLH.GT.TEM13) WRITE (7,4) 00002360
        IF (CLH.GT.TEM13) WRITE (7,8) CLH,TEM13 00002370
C                                00002380
C*****00002390
C* THE FOURTH ERROR IS PRINTED WHEN THE YEARLY CORRECTIVE MAINTENANCE *00002400
C* HOURS BY DEPOT PERSONNEL EXCEEDS THE TOTAL AMOUNT OF MAN-HOURS *00002410
C* AVAILABLE AT THE DEPOT. *00002420
C*****00002430
C                                00002440
        TEM14 = FLOAT(NDP)*DAA 00002450
        IF (DLH.GT.TEM14) WRITE (7,4) 00002460
        IF (DLH.GT.TEM14) WRITE (7,9) DLH,TEM14 00002470
C                                00002480
C                                00002490
222 RETURN 00002500
    END

```



```

SUBROUTINE OTAB2
C
C*****00000010
C*****00000020
C*****00000030
C* THIS SUBROUTINE PRINT OUTPUT TABLE 2:  OUTPUTS BY LRU TYPE.  *00000040
C*****00000050
C
COMMON /RD4/ INO(200),LRNAME(200,24),LEPART(200,12),NCI(200),
+ MTBI(200),N,PA(200),QPA(200),QR(200),RN(200),
+ UC(200)
INTEGER QPA,QR
REAL MTBI
C
COMMON /C2/ C2,C2C,C2D,C2L(200),C2S
C
COMMON /C3/ C3,C3C,C3D,C3L(200),C3S
C
COMMON /C4/ C4,C4I,C4L(200),C4R
C
COMMON /C9/ C9,C9C,C9D,C9I,C9L(200),C9R,C9S
C
2  FORMAT (1H1,19X,34HOUTPUT TABLE 2:  LRU COST ELEMENTS/31X,
+ 15H($ IN MILLIONS)///)
3  FORMAT (1H1,19X,46HOUTPUT TABLE 2:  LRU COST ELEMENTS (CONTINUED)/
+ 31X,15H($ IN MILLIONS)///)
4  FORMAT (31X,36HCORREC- INVEST- REPLEN- INVENTORY/
+ 4H LRU,6X,3HLRU,18X,4HTIVE,5X,4HMENT,5X,7HISHMENT,2X,
+ 7HMANAGE-/4H INO,6X,4HNOON,17X,6HMAINT.,3X,6HSPARES,
+ 3X,6HSPARES,3X,4HMENT,5X,5HTOTAL/)
5  FORMAT (1X,13,2X,24A1,F7.2,F9.2,F9.2,F9.2,F8.2)
6  FORMAT (1H0,11X,6HTOTAL:,12X,F7.2,F9.2,F9.2,F9.2,F8.2///
+ 5X,52HNOTE:  ONLY THOSE PORTIONS OF COST ELEMENTS DIRECTLY,
+ 7H ATTRI-/12X,45HNOTABLE TO INDIVIDUAL LRU TYPES ARE DISPLAYED,
+ 6H HERE.)
C
C
WRITE(7,2)
WRITE(7,4)
C
LINCNT = 0
TOT1 = 0.
TOT2 = 0.
TOT3 = 0.
TOT4 = 0.
DO 15 I=1,N
IF (QPA(I).EQ.0) GO TO 15
TOT = (C2L(I) + C3L(I) + C4L(I) + C9L(I)) / 1000000.
T1 = C2L(I) / 1000000.
T2 = C3L(I) / 1000000.
T3 = C4L(I) / 1000000.
T4 = C9L(I) / 1000000.
WRITE(7,5) I,(LRNAME(I,J),J=1,24),T1,T2,T3,T4,TOT
TOT1 = TOT1 + C2L(I)
TOT2 = TOT2 + C3L(I)
TOT3 = TOT3 + C4L(I)
TOT4 = TOT4 + C9L(I)
LINCNT = LINCNT + 1
IF (LINCNT.LT.50) GO TO 15
WRITE(7,3)
WRITE(7,4)
LINCNT = 0
15 CONTINUE
C
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550
00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630

```



```

IF (LINCNT.LE.45) GO TO 16
WRITE (7,3)
WRITE (7,4)
16 TOT = (TOT1 + TOT2 + TOT3 + TOT4) / 1000000.
TOT1 = TOT1 / 1000000.
TOT2 = TOT2 / 1000000.
TOT3 = TOT3 / 1000000.
TOT4 = TOT4 / 1000000.
WRITE (7,6) TOT1,TOT2,TOT3,TOT4,TOT
C
RETURN
END
SUBROUTINE OTAB3
C
C*****
C* SUBROUTINE OTAB3 PRINTS OUTPUT TABLE 3: OUTPUT BY LOCATION
C* (SITE, CMP, DEPOT).
C*****
C
COMMON /C2/ C2,C2C,C2D,C2L(200),C2S
C
COMMON /C3/ C3,C3C,C3D,C3L(200),C3S
C
COMMON /C4/ C4,C4I,C4L(200),C4R
C
COMMON /C5/ C5
C
COMMON /C6/ C6,C6C,C6D,C6I,C6R,C6S
C
COMMON /C7/ C7
C
COMMON /C8/ C8,C8C,C8D,C8I,C8R,C8S
C
COMMON /C9/ C9,C9C,C9D,C9I,C9L(200),C9R,C9S
C
COMMON /C10/ C10,C10HW(10),C10I,C10R,C10SW(10)
C
C
1 FORMAT (1H1,17X,42HOUTPUT TABLE 3: COST ELEMENTS BY LOCATION/
+ 31X,15H($ IN MILLIONS)////
+ 4X,12HCOST ELEMENT,12X,4HSITE,5X,3HCMP,5X,5HDEPOT,5X,5HTOTAL//
+ 12H MAINTENANCE,11X,3F9.2,F10.2/
+ 18H INVESTMENT SPARES,5X,3F9.2,F10.2/
+ 21H REPLENISHMENT SPARES,2X,2F9.2,9H 0.0 ,F10.2/
+ 12H CONSUMABLES,11X,F9.2,2(9H 0.0 ),F10.2/
+ 18H SUPPORT EQUIPMENT,5X,3F9.2,F10.2/
+ 11H FACILITIES,12X,F9.2,2(9H 0.0 ),F10.2/
+ 9H TRAINING,14X,3F9.2,F10.2/
+ 21H INVENTORY MANAGEMENT,2X,3F9.2,F10.2/
+ 21H SOFTWARE MAINTENANCE,2X,2(9H 0.0 ),F9.2,F10.2//
+ 5X,11HTOTAL COST:,7X,3F9.2,F10.2)
C
C
T2S = C2S / 1000000.
T2C = C2C / 1000000.
T2D = C2D / 1000000.
T2 = C2 / 1000000.
T3S = C3S / 1000000.
T3C = C3C / 1000000.
T3D = C3D / 1000000.
T3 = C3 / 1000000.
T4R = C4R / 1000000.
T4I = C4I / 1000000.

```

```

T4 = C4 / 1000000.
T5 = C5 / 1000000.
T6S = C6S / 1000000.
T6C = C6C / 1000000.
T6D = C6D / 1000000.
T6 = C6 / 1000000.
T7 = C7 / 1000000.
T8S = C8S / 1000000.
T8C = C8C / 1000000.
T8D = C8D / 1000000.
T8 = C8 / 1000000.
T9S = C9S / 1000000.
T9C = C9C / 1000000.
T9D = C9D / 1000000.
T9 = C9 / 1000000.
T10 = C10 / 1000000.
TOT1 = (C2S + C3S + C4R + C5 + C6S + C7 + C8S + C9S) / 1000000.
TOT2 = (C2C + C3C + C4I + C6C + C8C + C9C) / 1000000.
TOT3 = (C2D + C3D + C6D + C8D + C9D + C10) / 1000000.
TOT4 = TOT1 + TOT2 + TOT3
WRITE (7, 1) T2S, T2C, T2D, T2, T3S, T3C, T3D, T3, T4R, T4I, T4,
+ T5, T5, T6S, T6C, T6D, T6, T7, T7, T8S, T8C, T8D,
+ T8, T9S, T9C, T9D, T9, T10, T10, TOT1, TOT2, TOT3, TOT4
C
RETURN
END
SUBROUTINE OTAB4
C
C*****
C* SUBROUTINE THAT PRINTS OUTPUT TABLE 4: OUTPUTS BY FUNCTIONAL AREA.*
C*****
C
COMMON /RD3/ ENYR (10), PA, PANAME (10, 24), HWCF (10), INOPA (10), HHW (10),
+ SIZE (10), SHI (10), SWCF (10), SWFIX (10), SWVAR (10),
+ WEIGHT (10)
INTEGER PA
REAL HHW
C
COMMON /NCOS/ CLH, PHPA (10), SHMH, SPHH, STNH
C
COMMON /C10/ C10, C10HW (10), C10I, C10R, C10SW (10)
C
1 FORMAT (1H1, 17X, 43HOUTPUT TABLE 4: RESULTS BY FUNCTIONAL AREA////
+ 33X, 34HANNUAL SITE . ENHANCEMENT COST/
+ 33X, 33HPREVENTIVE . ($/LIFE CYCLE) /
+ 10X, 15HFUNCTIONAL AREA, 8X, 13HMAINTENANCE ./
+ 22H INDEX NOMENCLATURE, 11X, 28H(MAN-HOURS) . SOFTWARE HARD,
+ 11HWARE TOTAL/45X, 1H.)
2 FORMAT (1X, I3, 3X, 24A1, F10.2, 4X, 1H., F8.2, F9.2, F9.2)
3 FORMAT (45X, 1H./16X, 6HTOTAL: , 9X, F10.2, 4X, 1H., F8.2, F9.2, F9.2)
C
WRITE (7, 1)
TOT1 = 0.
TOT2 = 0.
TOT3 = 0.
DO 5 J=1, PA
T1 = C10SW(J) / 1000000.
T2 = C10HW(J) / 1000000.
TOT1 = TOT1 + PHPA(J)
TOT2 = TOT2 + T1
TOT3 = TOT3 + T2
TOT4 = T1 + T2

```

	WRITE(7,2) INOFA(J), (PANAME(J,L), L=1,24), PHFA(J), T1, T2, TOT4	00001900
5	CONTINUE	00001910
		00001920
C	TOT4 = TOT2 + TOT3	00001930
	WRITE(7,3) TOT1, TOT2, TOT3, TOT4	00001940
C		00001950
	RETURN	00001960
	END	00001970

```

SUBROUTINE OTAB5
C
C*****
C* SUBROUTINE OTAB5
C* SUBROUTINE THAT PRINTS OUTPUT TABLE 5: MISSION CRITICAL LRUS
C* NOT STOCKED AT SITE.
C*****
C
COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200),
+ MTBI(200),N,PA(200),QPA(200),QR(200),RH(200),
+ UC(200)
INTEGER QPA,QR
REAL MTBI
C
COMMON /AUX5/ DSTK(200),ECMT,HCF(200),STK(200),STK1(200),Z
INTEGER STK,STK1
C
1 FORMAT (1H1,6X,30HOUTPUT TABLE 5: MISSION CRITI,
+ 29HCAL LRUS NOT STOCKED ON SITE ///
+ 52X,6HNUMBER,5X,9HMEAN TIME/
+ 2X,3HLRU,47X,7HOF LRUS,4X,7HBETWEEN/
+ 1X,5HINDEX,46X,7HIN EACH,4X,6HMAINT./
+ 1X,6HNUMBER,5X,4HNAME,20X,11HPART NUMBER,5X,5HRADAR,6X,
+ 9HINCIDENTS/52X,5H(QPA),6X,6H(MTBI)/)
2 FORMAT (2X,I3,4X,24A1,3X,12A1,I8,5X,F11.3)
3 FORMAT (//8X,47H MISSION CRITICAL LRUS ARE ALL STOCKED ON SITE.)
4 FORMAT (///51H AVERAGE NUMBER OF EMERGENCY CORRECTIVE MAINTENANCE/
+ 44H TRIPS FROM CHP TO THE RADAR SITES PER YEAR:,12X,F10.3)
C
C WRITE(7,1)
C
C ICNT = 0
DO 5 I=1,N
IF (HCF(I).EQ.0.OR.QPA(I).EQ.0) GO TO 5
WRITE(7,2) I,(LRNAME(I,J),J=1,24),(LRPART(I,K),K=1,12),QPA(I),
+ MTBI(I)
ICNT = 1
5 CONTINUE
C
IF (ICNT.EQ.0) WRITE(7,3)
WRITE(7,4) ECMT
C
RETURN
END

```



```

SUBROUTINE OSENS
C
C*****
C* SUBROUTINE OSENS WRITES OUT AT EITHER THE TERMINAL OR THE OFF-LINE
C* PRINTER ANY NUMBER OF DERIVATIVES THE USER REQUESTS. THESE ARE
C* DISPLAYED IN DECREASING ORDER WITH RESPECT TO TOTAL CHANGE IN LCC
C* (IN MILLIONS OF DOLLARS). OUTPUT CAN BE SENT TO THE TERMINAL AND/
C* OR TO THE OFF-LINE PRINTER AT THE USER'S DISCRETION.
C*****
COMMON /RD4/ INO(200), LRNAME(200,24), LRPART(200,12), HCI(200),
+ MTBI(200), N, PA(200), QPA(200), QR(200), RM(200),
+ UC(200)
INTEGER QPA, QR
REAL MTBI
COMMON /RD5/ BCMH(200), CBCMH(200), CHH(200), CRTS1(200), CRTS2(200),
+ DBCMh(200), DHH(200), DRTS1(200), DRTS2(200),
+ DRTS3(200), PPR(200), INH(200), INO15(200), N15,
+ PAMH(200), RIP(200), RL(200), RMH(200), SMH(200), SRTS1(200),
+ WOR1(200), WOR2(200), WOR3(200)
REAL INH
INTEGER RL
COMMON /SENS/ FINC, LDCMH, LDCRTS, LDDMH, LDDRTS, LDERV, LDFPR, LDFR,
+ LDHCI, LDRH, LDSRTS, LDWOR, LDUC
COMMON /CNTL/ EXIT, ITER, MAXPMT, PRNT, REDO, REREAD
INTEGER EXIT, PRNT, REDO, REREAD
COMMON /TDXUC/ TDXUC
COMMON /TDFR/ IDFR(200), TDFR(200)
COMMON /TDK/ TDK
COMMON /TDUC/ IDUC(200), TDUC(200)
COMMON /TDWOR/ IDWOR(200), TDWOR(200), WF(200)
COMMON /TDFPR/ IDFPR(200), TDFPR(200)
COMMON /TDCMH/ IDC MH(200), TDCMH(200)
COMMON /TDDMH/ IDDMH(200), TDDMH(200)
COMMON /TDRH/ IDRH(200), TDRH(200)
COMMON /TDSRTS/ IDSRTS(200), SPSRTS(200), TDSRTS(200)
COMMON /TDCRTS/ CFCRTS(200), IDCRTS(200), TDCRTS(200)
COMMON /TDDRTS/ DFDRTS(200), IDDRTS(200), TDDRTS(200)
COMMON /TDHCI/ IDHCI(200), HCIC(200), TDHCI(200)
DIMENSION TEMP(200), ITEMP(200), TEMP2(200)
1 FORMAT (1H1,25X,26HSENSITIVITY ANALYSIS TABLE)
2 FORMAT (1H1,20X,37HOUTPUT TABLE 6: SENSITIVITY ANALYSIS//)
3 FORMAT (/23H CHANGE IN LCC DUE TO A,P5.1,15H % INCREASE IN://)

```

	* 36H GLOBAL EQUIPMENT COST (XUC) (\$M): ,P15.3/	00000640
	* 41H GLOBAL FAILURE RATE (K FACTOR) (\$M): ,P10.3)	00000650
C		00000660
4	FORMAT (/12H LRU INDEX,17X,6(4X,I3))	00000670
C		00000680
5	FORMAT (21H CHANGE IN LCC (\$M),9X,6P7.3)	00000690
C		00000700
6	FORMAT (/12H LRU INDEX,17X,12(4X,I3))	00000710
C		00000720
7	FORMAT (21H CHANGE IN LCC (\$M),9X,12P7.3)	00000730
C		00000740
8	FORMAT (///40H LRU UNIT COST (ORDERED BY SENSITIVITY):)	00000750
C		00000760
9	FORMAT (22H CHANGE IN UC ,8X,6(1X,P6.0))	00000770
C		00000780
10	FORMAT (22H CHANGE IN UC ,8X,12(1X,P6.0))	00000790
C		00000800
11	FORMAT (///43H LRU FAILURE RATE (ORDERED BY SENSITIVITY):)	00000810
C		00000820
12	FORMAT (29H CHANGE IN FR (PPH) ,1X,6(1X,P6.2))	00000830
C		00000840
13	FORMAT (29H CHANGE IN FR (PPH) ,1X,12(1X,P6.2))	00000850
C		00000860
14	FORMAT (///44H LRU WEAR-OUT RATE (ORDERED BY SENSITIVITY):)	00000870
C		00000880
15	FORMAT (26H CHANGE IN WOR ,4X,6(1X,P6.3))	00000890
C		00000900
16	FORMAT (26H CHANGE IN WOR ,4X,12(1X,P6.3))	00000910
C		00000920
17	FORMAT (///46H LRU FALSE PULL RATE (ORDERED BY SENSITIVITY):)	00000930
C		00000940
18	FORMAT (28H CHANGE IN PPR ,2X,6(1X,P6.3))	00000950
C		00000960
19	FORMAT (28H CHANGE IN PPR ,2X,12(1X,P6.3))	00000970
C		00000980
20	FORMAT (///46H LRU CMF REPAIR TIME (ORDERED BY SENSITIVITY):)	00000990
C		00001000
21	FORMAT (28H CHANGE IN CMH ,2X,6(1X,P6.2))	00001010
C		00001020
22	FORMAT (28H CHANGE IN CMH ,2X,12(1X,P6.2))	00001030
C		00001040
23	FORMAT (///48H LRU DEPOT REPAIR TIME (ORDERED BY SENSITIVITY):)	00001050
C		00001060
24	FORMAT (28H CHANGE IN DMH ,2X,6(1X,P6.2))	00001070
C		00001080
25	FORMAT (28H CHANGE IN DMH ,2X,12(1X,P6.2))	00001090
C		00001100
26	FORMAT (///46H LRU REPAIR MATERIAL (ORDERED BY SENSITIVITY):)	00001110
C		00001120
27	FORMAT (28H CHANGE IN RM ,2X,6(1X,P6.0))	00001130
C		00001140
28	FORMAT (28H CHANGE IN RM ,2X,12(1X,P6.0))	00001150
C		00001160
29	FORMAT (///51H LRU SITE REPAIR FRACTION (ORDERED BY SENSITIVITY, ,	00001170
	* 18HNOT INCLUDING SE):)	00001180
C		00001190
30	FORMAT (28H CHANGE IN SRTS ,2X,6(1X,P6.3))	00001200
C		00001210
31	FORMAT (28H CHANGE IN SRTS ,2X,12(1X,P6.3))	00001220
C		00001230
32	FORMAT (///50H LRU CMF REPAIR FRACTION (ORDERED BY SENSITIVITY, ,	00001240
	* 18HNOT INCLUDING SE):)	00001250
C		00001260

```

33  FORMAT (27H  CHANGE IN CRTS          ,3X,6(1X,F6.3))          00001270
C                                     00001280
34  FORMAT (27H  CHANGE IN CRTS          ,3X,12(1X,F6.3))         00001290
C                                     00001300
35  FORMAT(///52H LRU DEPOT REPAIR FRACTION (ORDERED BY SENSITIVITY, ,00001310
+ 18HNOT INCLUDING SE):)          00001320
C                                     00001330
36  FORMAT (28H  CHANGE IN DRTS          ,2X,6(1X,F6.3))          00001340
C                                     00001350
37  FORMAT (28H  CHANGE IN DRTS          ,2X,12(1X,F6.3))         00001360
C                                     00001370
38  FORMAT (///47H LRU MISSION CRITICAL (ORDERED BY SENSITIVITY, ,00001380
+ 24HFOR LRUS WITH MCI >= 1):)    00001390
C                                     00001400
39  FORMAT (16H  CHANGE IN MCI,13X,6(4X,I3))          00001410
C                                     00001420
40  FORMAT (16H  CHANGE IN MCI,13X,12(4X,I3))          00001430
C                                     00001440
C*****00001450
C*  PRINT OUT HEADINGS AT THE TERMINAL AND/OR AT THE OFF-LINE      *00001460
C*  PRINTER.                                                        *00001470
C*****00001480
C                                     00001490
      IF (PRNT.NE.1) WRITE (6,1)
      IF (PRNT.NE.0) WRITE (7,2)
C                                     00001500
C                                     00001510
C                                     00001520
      TEM1 = FINC * 100
      TEM2 = TDXUC / 1000000.
      TEM3 = TDK / 1000000.
      IF (PRNT.NE.1) WRITE (6,3) TEM1,TEM2,TEM3
      IF (PRNT.NE.0) WRITE (7,3) TEM1,TEM2,TEM3
C                                     00001530
C                                     00001540
C                                     00001550
C                                     00001560
C*****00001570
C*  THIS SECTION OF CODE (FROM HERE UNTIL LABELL NUMBER 22001) IS RE- *00001580
C*  PEATED 11 TIMES -- ONCE FOR EACH DERIVATIVE. AT THE START OF EACH *00001590
C*  SECTION, VARIABLES (LDERV AND LDUC) ARE CHECKED TO SEE IF ANY *00001600
C*  DERIVATIVES OF THIS TYPE ARE REQUESTED. IF NOT, CONTROL IS *00001610
C*  TRANSFERED TO THE NEXT SECTION. EACH OF THE 11 SECTIONS IS DIVI- *00001620
C*  DED INTO 2 PARTS - ONE FOR PRINTING AT THE TERMINAL AND THE OTHER *00001630
C*  FOR THE OFF-LINE PRINTER. *00001640
C*****00001650
C                                     00001660
C*****00001670
C*  THE FIRST SECTION COMPUTES LRU UNIT COST ORDERED BY SENSITIVITY *00001680
C*****00001690
C                                     00001700
      IF (LDUC.EQ.0.AND.LDERV.EQ.0) GO TO 2001
C                                     00001710
C                                     00001720
C                                     00001730
      ILDUC = MAX0(LDERV,LDUC)
      DO 50 I=1,ILDUC
      TEMP(I) = FINC * UC(IDUC(I))
      TEMP2(I) = TDUC(IDUC(I)) / 1000000.
50  CONTINUE
C                                     00001740
C                                     00001750
C                                     00001760
C                                     00001770
C                                     00001780
C                                     00001790
      IF (LDUC.EQ.0.OR.PRNT.EQ.1) GO TO 1001
      WRITE (6,8)
      IFIN = 0
      IBEG = 1
      INUM = 0
      ILDUC = LDUC
C                                     00001800
C                                     00001810
C                                     00001820
C                                     00001830
C                                     00001840
C                                     00001850
C*****00001860
C*  COMPUTE CHANGE IN UNIT COST FOR NUMBER OF DERIVATIVES REQUESTED. *00001870

```



```

C*****00001880
C
DO 1000 I=1,34                                00001890
    IF (ILDUC.GT.6) GO TO 100                    00001900
    INUM = ILDUC + INUM                          00001910
    IFIN = 1                                     00001920
    GO TO 150                                    00001930
100    INUM = 6 + INUM                           00001940
    ILDUC = ILDUC-6                             00001950
C                                                00001960
C                                                00001970
C*****00001980
C PRINT OUT DERIVATIVES AT THE TERMINAL (6 PER LINE) *00001990
C*****00002000
C
150    WRITE (6,4) (IDUC(J),J=IBEG,INUM)         00002010
    WRITE (6,9) (TEMP(J),J=IBEG,INUM)           00002020
    WRITE (6,5) (TEMP2(J),J=IBEG,INUM)          00002030
    IBEG = INUM + 1                             00002040
C                                                00002050
C                                                00002060
    IF (IFIN.EQ.1) GO TO 1001                   00002070
1000    CONTINUE                                00002080
C                                                00002090
C*****00002100
C* IF PRNT=0 THEN DON'T SEND THE OUTPUT TO THE OFF-LINE PRINTER. *00002110
C*****00002120
C
1001    IF (PRNT.EQ.0) GO TO 2001                00002130
    WRITE (7,8)                                  00002140
    IFIN = 0                                     00002150
    IBEG = 1                                     00002160
    INUM = 0                                     00002170
C                                                00002180
C                                                00002190
C*****00002200
C* LDERV IS THE MINIMUM NUMBER OF DERIVATIVES PRINTED AT THE OFF-LINE *00002210
C* PRINTER FOR AN LRU. E.G., IF THE USER REQUESTS THAT 5 DERIVATIVES *00002220
C* BE PRINTED AND LDERV IS 7, THEN 7 DERIVATIVES WILL BE PRINTED FOR * 00002230
C* THAT LRU. *00002240
C*****00002250
C
    ILDUC = MAX0(LDERV,LDUC)                     00002260
C
DO 2000 I=1,17                                00002270
C
    IF (ILDUC.GT.12) GO TO 1100                  00002280
    INUM = ILDUC + INUM                          00002290
    IFIN = 1                                     00002300
    GO TO 1150                                    00002310
1100    INUM = 12 + INUM                         00002320
    ILDUC = ILDUC-12                             00002330
C                                                00002340
C                                                00002350
C                                                00002360
C                                                00002370
C*****00002380
C* PRINT OUT DERIVATIVES AT THE OFF-LINE PRINTER. (12 PER LINE) *00002390
C*****00002400
C
1150    WRITE (7,6) (IDUC(J),J=IBEG,INUM)         00002410
    WRITE (7,10) (TEMP(J),J=IBEG,INUM)           00002420
    WRITE (7,7) (TEMP2(J),J=IBEG,INUM)          00002430
    IBEG = INUM + 1                             00002440
C                                                00002450
C                                                00002460
    IF (IFIN.EQ.1) GO TO 2001                   00002470
2000    CONTINUE                                00002480
C                                                00002490
C*****00002500

```



```

C* THE SECOND SECTION PRINTS OUT LRU FAILURE RATE ORDERED BY SENSI- *00002510
C* TIVITY. *00002520
C***** *00002530
C *00002540
2001 IF (LDPR.EQ.0.AND.LDERV.EQ.0) GO TO 4001 *00002550
C *00002560
    ILDFR = MAX0(LDERV,LDPR) *00002570
    DO 2050 I=1,ILDFR *00002580
        TEMP(I) = FINC * (10E6/MTBI(IDFR(I))) *00002590
        TEMP2(I) = TDFR(IDFR(I)) / 1000000. *00002600
2050 CONTINUE *00002610
C *00002620
    IF (LDPR.EQ.0.OR.PRNT.EQ.1) GO TO 3001 *00002630
    WRITE (6,11) *00002640
    IPIN = 0 *00002650
    IBEG = 1 *00002660
    INUM = 0 *00002670
    ILDFR = LDPR *00002680
C *00002690
C***** *00002700
C* COMPUTE CHANGE IN FAILURE RATE IN PARTS PER MILLION FOR NUMBER OF *00002710
C* DERIVATIVES REQUESTED. *00002720
C***** *00002730
C *00002740
    DO 3000 I=1,34 *00002750
        IF (ILDFR.GT.6) GO TO 2100 *00002760
        INUM = ILDFR + INUM *00002770
        IPIN = 1 *00002780
        GO TO 2150 *00002790
2100 INUM = 6 + INUM *00002800
    ILDFR = ILDFR-6 *00002810
C *00002820
C***** *00002830
C* PRINT OUT DERIVATIVES AT TERMINAL (6 PER LINE). *00002840
C***** *00002850
C *00002860
2150 WRITE (6,4) (IDFR(J),J=IBEG,INUM) *00002870
    WRITE (6,12) (TEMP(J),J=IBEG,INUM) *00002880
    WRITE (6,5) (TEMP2(J),J=IBEG,INUM) *00002890
    IBEG = INUM + 1 *00002900
C *00002910
    IF (IPIN.EQ.1) GO TO 3001 *00002920
3000 CONTINUE *00002930
C *00002940
3001 IF (PRNT.EQ.0) GO TO 4001 *00002950
    WRITE (7,11) *00002960
    IPIN = 0 *00002970
    IBEG = 1 *00002980
    INUM = 0 *00002990
    ILDFR = MAX0(LDERV,LDPR) *00003000
C *00003010
    DO 4000 I=1,17 *00003020
C *00003030
        IF (ILDFR.GT.12) GO TO 3100 *00003040
        INUM = ILDFR + INUM *00003050
        IPIN = 1 *00003060
        GO TO 3150 *00003070
3100 INUM = 12 + INUM *00003080
    ILDFR = ILDFR-12 *00003090
C *00003100
C***** *00003110
C* PRINT OUT DERIVATIVES AT THE OFF-LINE PRINTER (12 PER LINE). *00003120
C***** *00003130

```

C		00003140
3150	WRITE (7,6) (IDFR(J),J=IBEG,INUM)	00003150
	WRITE (7,13) (TEMP(J),J=IBEG,INUM)	00003160
	WRITE (7,7) (TEMP2(J),J=IBEG,INUM)	00003170
	IBEG = INUM + 1	00003180
C		00003190
	IF (IPIN.EQ.1) GO TO 4001	00003200
4000	CONTINUE	00003210
C		00003220
C*****		00003230
C*	THE THIRD SECTION PRINTS OUT LRU WEAR-OUT RATE ORDERED BY	*00003240
C*	SENSITIVITY.	*00003250
C*****		00003260
C		00003270
4001	IF (LDWOR.EQ.0.AND.LDERV.EQ.0) GO TO 6001	00003280
C		00003290
	ILDWOR = MAX0(LDERV,LDWOR)	00003300
	DO 4050 I=1,ILDWOR	00003310
	TEMP(I) = WF(IDWOR(I))	00003320
	TEMP2(I) = TDWOR(IDWOR(I)) / 1000000.	00003330
4050	CONTINUE	00003340
C		00003350
	IF (LDWOR.EQ.0.OR.PRNT.EQ.1) GO TO 5001	00003360
	WRITE (6,14)	00003370
	IPIN = 0	00003380
	IBEG = 1	00003390
	INUM = 0	00003400
	ILDWOR = LDWOR	00003410
C		00003420
C*****		00003430
C*	COMPUTE CHANGE IN WEAR-OUT RATE FOR NUMBER OF DERIVATIVES REQUESTED	*00003440
C*****		00003450
C		00003460
	DO 5000 I=1,34	00003470
	IF (ILDWOR.GT.6) GO TO 4100	00003480
	INUM = ILDWOR + INUM	00003490
	IPIN = 1	00003500
	GO TO 4150	00003510
4100	INUM = 6 + INUM	00003520
	ILDWOR = ILDWOR-6	00003530
C		00003540
C*****		00003550
C*	PRINT OUT DERIVATIVES AT TERMINAL. (6 PER LINE)	*00003560
C*****		00003570
C		00003580
4150	WRITE (6,4) (IDWOR(J),J=IBEG,INUM)	00003590
	WRITE (6,15) (TEMP(J),J=IBEG,INUM)	00003600
	WRITE (6,5) (TEMP2(J),J=IBEG,INUM)	00003610
	IBEG = INUM + 1	00003620
C		00003630
	IF (IPIN.EQ.1) GO TO 5001	00003640
5000	CONTINUE	00003650
C		00003660
5001	IF (PRNT.EQ.0) GO TO 6001	00003670
	WRITE (7,14)	00003680
	IPIN = 0	00003690
	IBEG = 1	00003700
	INUM = 0	00003710
	ILDWOR = MAX0(LDERV,LDWOR)	00003720
C		00003730
	DO 6000 I=1,17	00003740
C		00003750
	IF (ILDWOR.GT.12) GO TO 5100	00003760

```

      INUM = ILDWOR + INUM
      IPIN = 1
      GO TO 5150
5100  INUM = 12 + INUM
      ILDWOR = ILDWOR-12
C
C*****
C* PRINT OUT DERIVATIVES AT THE OFF-LINE PRINTER (12 PER LINE)
C*****
C
5150  WRITE (7,6) (IDWOR(J),J=IBEG,INUM)
      WRITE (7,16) (TEMP(J),J=IBEG,INUM)
      WRITE (7,7) (TEMP2(J),J=IBEG,INUM)
      IBEG = INUM + 1
C
      IF (IPIN.EQ.1) GO TO 6001
6000  CONTINUE
C
C*****
C* THE FOURTH SECTION PRINTS OUT LRU FALSE PULL RATE ORDERED BY
C* SENSITIVITY.
C*****
6001  IF (LDFPR.EQ.0.AND.LDERV.EQ.0) GO TO 8001
C
      ILDFPR = MAX0(LDERV,LDFPR)
      DO 6050 I=1,ILDFPR
        TEMP(I) = FINC * FPR(IDFPR(I))
        TEMP2(I) = TDFPR(IDFPR(I)) / 1000000.
6050  CONTINUE
C
      IF (LDFPR.EQ.0.OR.PRNT.EQ.1) GO TO 7001
      WRITE (6,17)
      IPIN = 0
      IBEG = 1
      INUM = 0
      ILDFPR = LDFPR
C
C*****
C* COMPUTE CHANGE IN FALSE PULL RATE FOR NUMBER OF DERIVATIVES
C* REQUESTED.
C*****
C
      DO 7000 I=1,34
        IF (ILDFPR.GT.6) GO TO 6100
        INUM = ILDFPR + INUM
        IPIN = 1
        GO TO 6150
6100  INUM = 6 + INUM
      ILDFPR = ILDFPR-6
C
C*****
C* PRINT OUT DERIVATIVES AT TERMINAL (6 PER LINE)
C*****
C
6150  WRITE (6,4) (IDFPR(J),J=IBEG,INUM)
      WRITE (6,18) (TEMP(J),J=IBEG,INUM)
      WRITE (6,5) (TEMP2(J),J=IBEG,INUM)
      IBEG = INUM + 1
C
      IF (IPIN.EQ.1) GO TO 7001
7000  CONTINUE
C

```



```

7001 IF (PRNT.EQ.0) GO TO 8001                                00004400
      WRITE (7,17)                                           00004410
      IPIN = 0                                               00004420
      IBEG = 1                                               00004430
      INUM = 0                                               00004440
      ILDFPR = MAXO(LDERV,LDFFPR)                            00004450
C                                           00004460
      DO 8000 I=1,17                                         00004470
C                                           00004480
      IF (ILDFPR.GT.12) GO TO 7100                          00004490
      INUM = ILDFPR + INUM                                   00004500
      IPIN = 1                                               00004510
      GO TO 7150                                             00004520
7100 INUM = 12 + INUM                                       00004530
      ILDFPR = ILDFPR-12                                     00004540
C                                           00004550
C*****00004560
C* PRINT OUT DERIVATIVES AT THE OFF-LINE PRINTER (12 PER LINE) *00004570
C*****00004580
C                                           00004590
7150 WRITE (7,6) (IDFPR(J),J=IBEG,INUM)                   00004600
      WRITE (7,19) (TEMP(J),J=IBEG,INUM)                   00004610
      WRITE (7,7) (TEMP2(J),J=IBEG,INUM)                   00004620
      IBEG = INUM + 1                                       00004630
C                                           00004640
      IF (IPIN.EQ.1) GO TO 8001                             00004650
8000 CONTINUE                                              00004660
C                                           00004670
C*****00004680
C* THE FIFTH SECTION PRINTS OUT LRU CHF REPAIR TIME ORDERED BY *00004690
C* SENSITIVITY.                                           00004700
C*****00004710
C                                           00004720
8001 IF (LDCNH.EQ.0.AND.LDERV.EQ.0) GO TO 10001           00004730
C                                           00004740
      ILDCNH = MAXO(LDERV,LDCNH)                            00004750
      DO 8050 I=1,ILDCNH                                    00004760
      TEMP(I) = PINC * CNH(IDCNH(I)) -                     00004770
      TEMP2(I) = TDCNH(IDCNH(I)) / 1000000.                00004780
8050 CONTINUE                                              00004790
C                                           00004800
      IF (LDCNH.EQ.0.OR.PRNT.EQ.1) GO TO 9001              00004810
      WRITE (6,20)                                           00004820
      IPIN = 0                                               00004830
      IBEG = 1                                               00004840
      INUM = 0                                               00004850
      ILDCNH = LDCNH                                         00004860
C                                           00004870
C*****00004880
C* COMPUTE CHANGE IN CHF REPAIR TIME FOR NUMBER OF DERIVATIVES *00004890
C* REQUESTED.                                           00004900
C*****00004910
C                                           00004920
      DO 9000 I=1,34                                         00004930
      IF (ILDCNH.GT.6) GO TO 8100                          00004940
      INUM = ILDCNH + INUM                                   00004950
      IPIN = 1                                               00004960
      GO TO 8150                                             00004970
8100 INUM = 6 + INUM                                       00004980
      ILDCNH = ILDCNH-6                                     00004990
C                                           00005000
C*****00005010
C* PRINT OUT DERIVATIVES AT TERMINAL (6 PER LINE)         00005020

```



```

C*****00005030
C00005040
8150 WRITE (6,4) (IDCMH(J),J=IBEG,INUM)00005050
      WRITE (6,21) (TEMP(J),J=IBEG,INUM)00005060
      WRITE (6,5) (TEMP2(J),J=IBEG,INUM)00005070
      IBEG = INUM + 100005080
C00005090
      IF (IPIN.EQ.1) GO TO 900100005100
9000 CONTINUE00005110
C00005120
9001 IF (PRNT.EQ.0) GO TO 1000100005130
      WRITE (7,20)00005140
      IPIN = 000005150
      IBEG = 100005160
      INUM = 000005170
      ILDCMH = MAX0(LDERV,LDCHH)00005180
C00005190
      DO 10000 I=1,1700005200
C00005210
      IF (ILDCMH.GT.12) GO TO 91000005220
      INUM = ILDCMH + INUM00005230
      IPIN = 100005240
      GO TO 915000005250
9100 INUM = 12 + INUM00005260
      ILDCMH = ILDCMH-1200005270
C00005280
C*****00005290
C* PRINT OUT DERIVATIVES AT THE OFF-LINE PRINTER (12 PER LINE) *00005300
C*****00005310
C00005320
9150 WRITE (7,6) (IDCMH(J),J=IBEG,INUM)00005330
      WRITE (7,22) (TEMP(J),J=IBEG,INUM)00005340
      WRITE (7,7) (TEMP2(J),J=IBEG,INUM)00005350
      IBEG = INUM + 100005360
C00005370
      IF (IPIN.EQ.1) GO TO 1000100005380
10000 CONTINUE00005390
C00005400
C*****00005410
C* THE SIXTH SECTION PRINTS OUT LRU DEPOT REPAIR TIME ORDERED BY *00005420
C* SENSITIVITY. *00005430
C*****00005440
C00005450
10001 IF (LDDMH.EQ.0.AND.LDERV.EQ.0) GO TO 1200100005460
C00005470
      ILDDMH = MAX0(LDERV,LDDMH)00005480
      DO 10050 I=1,ILDDMH00005490
      TEMP(I) = FINC * DMH(ILDDMH(I))00005500
      TEMP2(I) = TDDMH(ILDDMH(I)) / 1000000.00005510
10050 CONTINUE00005520
C00005530
      IF (LDDMH.EQ.0.OR.PRNT.EQ.1) GO TO 1100100005540
      WRITE (6,23)00005550
      IPIN = 000005560
      IBEG = 100005570
      INUM = 000005580
      ILDDMH = LDDMH00005590
C00005600
C*****00005610
C* COMPUTE CHANGE IN DEPOT REPAIR TIME FOR NUMBER OF DERIVATIVES *00005620
C* REQUESTED. *00005630
C*****00005640
C00005650

```

```

DO 11000 I=1,34                                00005660
  IF (ILDDMH.GT.6) GO TO 10100                  00005670
  INUM = ILDDMH + INUM                          00005680
  IFIN = 1                                      00005690
  GO TO 10150                                  00005700
10100 INUM = 6 + INUM                          00005710
      ILDDMH = ILDDMH-6                        00005720
C                                              00005730
C*****00005740
C* PRINT OUT DERIVATIVES AT TERMINAL (6 PER LINE) *00005750
C*****00005760
C                                              00005770
10150 WRITE (6,4) (IDDMH(J),J=IBEG,INUM)      00005780
      WRITE (6,24) (TEMP(J),J=IBEG,INUM)      00005790
      WRITE (6,5) (TEMP2(J),J=IBEG,INUM)      00005800
      IBEG = INUM + 1                         00005810
C                                              00005820
      IF (IFIN.EQ.1) GO TO 11001              00005830
11000 CONTINUE                                00005840
C                                              00005850
11001 IF (PRNT.EQ.0) GO TO 12001              00005860
      WRITE (7,23)                            00005870
      IFIN = 0                                00005880
      IBEG = 1                                00005890
      INUM = 0                                00005900
      ILDDMH = MAX0(LDERV,ILDDMH)             00005910
C                                              00005920
DO 12000 I=1,17                              00005930
C                                              00005940
      IF (ILDDMH.GT.12) GO TO 11100           00005950
      INUM = ILDDMH + INUM                    00005960
      IFIN = 1                                00005970
      GO TO 11150                             00005980
11100 INUM = 12 + INUM                         00005990
      ILDDMH = ILDDMH-12                     00006000
C                                              00006010
C*****00006020
C* PRINT OUT DERIVATIVES AT THE OFF-LINE PRINTER (12 PER LINE) *00006030
C*****00006040
C                                              00006050
11150 WRITE (7,6) (IDDMH(J),J=IBEG,INUM)      00006060
      WRITE (7,25) (TEMP(J),J=IBEG,INUM)      00006070
      WRITE (7,7) (TEMP2(J),J=IBEG,INUM)      00006080
      IBEG = INUM + 1                         00006090
C                                              00006100
      IF (IFIN.EQ.1) GO TO 12001              00006110
12000 CONTINUE                                00006120
C                                              00006130
C*****00006140
C* THE SEVENTH SECTION PRINTS OUT LRU REPAIR MATERIAL COSTS ORDERED *00006150
C* BY SENSITIVITY.                            *00006160
C*****00006170
C                                              00006180
12001 IF (LDRM.EQ.0.AND.LDERV.EQ.0) GO TO 14001 00006190
C                                              00006200
      LDRM = MAX0(LDERV,LDRM)                 00006210
DO 12050 I=1,LDRM                             00006220
  TEMP(I) = FINC * RM(IDRM(I)) * UC(IDRM(I)) 00006230
  TEMP2(I) = TDRM(IDRM(I)) / 1000000.        00006240
12050 CONTINUE                                00006250
C                                              00006260
      IF (LDRM.EQ.0.OR.PRNT.EQ.1) GO TO 13001 00006270
      WRITE (6,26)                            00006280

```

IPIN = 0	00006290
IBEG = 1	00006300
INUM = 0	00006310
ILDRM = LDRM	00006320
C	00006330
C*****	00006340
C* COMPUTE CHANGE IN REPAIR MATERIAL COSTS FOR NUMBER OF DERIVATIVES	00006350
C* REQUESTED.	00006360
C*****	00006370
C	00006380
DO 13000 I=1,34	00006390
IF (ILDRM.GT.6) GO TO 12100	00006400
INUM = ILDRM + INUM	00006410
IPIN = 1	00006420
GO TO 12150	00006430
12100 INUM = 6 + INUM	00006440
ILDRM = ILDRM-6	00006450
C	00006460
C*****	00006470
C* PRINT OUT DERIVATIVES AT TERMINAL (6 PER LINE)	00006480
C*****	00006490
C	00006500
12150 WRITE (6,4) (IDRM(J),J=IBEG,INUM)	00006510
WRITE (6,27) (TEMP(J),J=IBEG,INUM)	00006520
WRITE (6,5) (TEMP2(J),J=IBEG,INUM)	00006530
IBEG = INUM + 1	00006540
C	00006550
IF (IPIN.EQ.1) GO TO 13001	00006560
13000 CONTINUE	00006570
C	00006580
13001 IF (PRNT.EQ.0) GO TO 14001	00006590
WRITE (7,26)	00006600
IPIN = 0	00006610
IBEG = 1	00006620
INUM = 0	00006630
ILDRM = MAX0(LDERV,LDRM)	00006640
C	00006650
DO 14000 I=1,17	00006660
C	00006670
IF (ILDRM.GT.12) GO TO 13100	00006680
INUM = ILDRM + INUM	00006690
IPIN = 1	00006700
GO TO 13150	00006710
13100 INUM = 12 + INUM	00006720
ILDRM = ILDRM-12	00006730
C	00006740
C*****	00006750
C* PRINT OUT DERIVATIVES AT THE OFF-LINE PRINTER (12 PER LINE)	00006760
C*****	00006770
C	00006780
13150 WRITE (7,6) (IDRM(J),J=IBEG,INUM)	00006790
WRITE (7,28) (TEMP(J),J=IBEG,INUM)	00006800
WRITE (7,7) (TEMP2(J),J=IBEG,INUM)	00006810
IBEG = INUM + 1	00006820
C	00006830
IF (IPIN.EQ.1) GO TO 14001	00006840
14000 CONTINUE	00006850
C	00006860
C*****	00006870
C* THE EIGHTH SECTION PRINTS OUT LRU SITE REPAIR FRACTION ORDERED BY	00006880
C* BY SENSITIVITY, NOT INCLUDING SE.	00006890
C*****	00006900
C	00006910


```

14001 IF (LDSRTS.EQ.0.AND.LDERV.EQ.0) GO TO 16001      00006920
C                                                         00006930
      ILDSRT = MAX0(LDERV,LDSRTS)                        00006940
      DO 14050 I=1,ILDSRT                               00006950
        TEMP(I) = SFSRTS(IDSRTS(I))                     00006960
        TEMP2(I) = TDSRTS(IDSRTS(I)) / 1000000.         00006970
14050 CONTINUE                                           00006980
C                                                         00006990
      IF (LDSRTS.EQ.0.OR.PRNT.EQ.1) GO TO 15001         00007000
      WRITE (6,29)                                       00007010
      IFIN = 0                                           00007020
      IBEG = 1                                           00007030
      INUM = 0                                           00007040
      ILDSRT = LDSRTS                                   00007050
C                                                         00007060
C*****00007070
C* COMPUTE CHANGE IN SITE REPAIR FRACTION FOR NUMBER OF DERIVATIVES *00007080
C* REQUESTED.                                           *00007090
C*****00007100
C                                                         00007110
      DO 15000 I=1,34                                    00007120
        IF (ILDSRT.GT.6) GO TO 14100                     00007130
        INUM = ILDSRT + INUM                             00007140
        IFIN = 1                                          00007150
        GO TO 14150                                       00007160
14100 INUM = 6 + INUM                                    00007170
      ILDSRT = ILDSRT-6                                  00007180
C                                                         00007190
C*****00007200
C* PRINT OUT DERIVATIVES AT TERMINAL (6 PER LINE)      *00007210
C*****00007220
C                                                         00007230
14150 WRITE (6,4) (IDSRTS(J),J=IBEG,INUM)              00007240
      WRITE (6,30) (TEMP(J),J=IBEG,INUM)               00007250
      WRITE (6,5) (TEMP2(J),J=IBEG,INUM)               00007260
      IBEG = INUM + 1                                    00007270
C                                                         00007280
C                                                         00007290
      IF (IFIN.EQ.1) GO TO 15001                         00007300
15000 CONTINUE                                           00007310
C                                                         00007320
15001 IF (PRNT.EQ.0) GO TO 16001                        00007330
      WRITE (7,29)                                       00007340
      IFIN = 0                                           00007350
      IBEG = 1                                           00007360
      INUM = 0                                           00007370
      ILDSRT = MAX0(LDERV,LDSRTS)                       00007380
C                                                         00007390
      DO 16000 I=1,17                                    00007400
C                                                         00007410
        IF (ILDSRT.GT.12) GO TO 15100                   00007420
        INUM = ILDSRT + INUM                             00007430
        IFIN = 1                                          00007440
        GO TO 15150                                       00007450
15100 INUM = 12 + INUM                                   00007460
      ILDSRT = ILDSRT-12                                 00007470
C                                                         00007480
C*****00007480
C* PRINT OUT DERIVATIVES AT THE OFF-LINE PRINTER (12 PER LINE) *00007490
C*****00007500
C                                                         00007510
15150 WRITE (7,6) (IDSRTS(J),J=IBEG,INUM)             00007520
      WRITE (7,31) (TEMP(J),J=IBEG,INUM)              00007530
      WRITE (7,7) (TEMP2(J),J=IBEG,INUM)              00007540

```



```

      IBEG = INUM + 1                                00007550
C                                                    00007560
      IF (IPIN.EQ.1) GO TO 16001                    00007570
16000 CONTINUE                                     00007580
C                                                    00007590
C*****00007600
C* THE NINTH SECTION PRINTS OUT LRU CMP REPAIR FRACTION ORDERED BY *00007610
C* SENSITIVITY, NOT INCLUDING SE.                  *00007620
C*****00007630
C                                                    00007640
16001 IF (LDCRTS.EQ.0.AND.LDERV.EQ.0) GO TO 18001 00007650
C                                                    00007660
      ILDCRT = MAX0(LDERV,LDCRTS)                   00007670
      DO 16050 I=1,ILDCRT                          00007680
        TEMP(I) = CPCRTS(IDCRTS(I))                 00007690
        TEMP2(I) = TDCRTS(IDCRTS(I)) / 1000000.     00007700
16050 CONTINUE                                     00007710
C                                                    00007720
      IF (LDCRTS.EQ.0.OR.PRNT.EQ.1) GO TO 17001    00007730
      WRITE (6,32)                                  00007740
      IPIN = 0                                       00007750
      IBEG = 1                                       00007760
      INUM = 0                                       00007770
      ILDCRT = LDCRTS                               00007780
C                                                    00007790
C*****00007800
C* COMPUTE CHANGE IN CMP REPAIR FRACTION FOR NUMBER OF DERIVATIVES *00007810
C* REQUESTED.                                       *00007820
C*****00007830
C                                                    00007840
      DO 17000 I=1,34                               00007850
        IF (ILDCRT.GT.6) GO TO 16100               00007860
        INUM = ILDCRT + INUM                       00007870
        IPIN = 1                                    00007880
        GO TO 16150                                 00007890
16100 INUM = 6 + INUM                               00007900
      ILDCRT = ILDCRT-6                             00007910
C                                                    00007920
C*****00007930
C* PRINT OUT DERIVATIVES AT TERMINAL (6 PER LINE) *00007940
C*****00007950
C                                                    00007960
16150 WRITE (6,4) (IDCRTS(J),J=IBEG,INUM)         00007970
      WRITE (6,33) (TEMP(J),J=IBEG,INUM)          00007980
      WRITE (6,5) (TEMP2(J),J=IBEG,INUM)          00007990
      IBEG = INUM + 1                              00008000
C                                                    00008010
      IF (IPIN.EQ.1) GO TO 17001                   00008020
17000 CONTINUE                                     00008030
C                                                    00008040
17001 IF (PRNT.EQ.0) GO TO 18001                   00008050
      WRITE (7,32)                                  00008060
      IPIN = 0                                       00008070
      IBEG = 1                                       00008080
      INUM = 0                                       00008090
      ILDCRT = MAX0(LDERV,LDCRTS)                 00008100
C                                                    00008110
      DO 18000 I=1,17                              00008120
C                                                    00008130
        IF (ILDCRT.GT.12) GO TO 17100              00008140
        INUM = ILDCRT + INUM                       00008150
        IPIN = 1                                    00008160
        GO TO 17150                                 00008170

```

```

17100  INUM = 12 + INUM          00008180
      ILDCRT = ILDCRT-12        00008190
C                                          00008200
C*****                                00008210
C* PRINT OUT DERIVATIVES AT THE OFF-LINE PRINTER (12 PER LINE) *00008220
C*****                                00008230
C                                          00008240
17150  WRITE (7,6) (IDCRTS(J),J=IBEG,INUM) 00008250
      WRITE (7,34) (TEMP(J),J=IBEG,INUM) 00008260
      WRITE (7,7) (TEMP2(J),J=IBEG,INUM) 00008270
      IBEG = INUM + 1              00008280
C                                          00008290
      IF (IPIN.EQ.1) GO TO 18001 00008300
18000 CONTINUE                    00008310
C                                          00008320
C*****                                00008330
C* THE TENTH SECTION PRINTS OUT LRU DEPOT REPAIR FRACTION ORDERED BY *00008340
C* SENSITIVITY, NOT INCLUDING SE. *00008350
C*****                                00008360
C                                          00008370
18001 IF (LDDRTS.EQ.0.AND.LDERV.EQ.0) GO TO 20001 00008380
C                                          00008390
      ILDDRT = MAX0(LDERV,LDDRTS) 00008400
      DO 18050 I=1,ILDDRT         00008410
      TEMP(I) = DFDRTS(IDDRTS(I)) 00008420
      TEMP2(I) = TDDRTS(IDDRTS(I)) / 1000000. 00008430
18050 CONTINUE                    00008440
C                                          00008450
      IF (LDDRTS.EQ.0.OR.PRNT.EQ.1) GO TO 19001 00008460
      WRITE (6,35)                 00008470
      IPIN = 0                     00008480
      IBEG = 1                     00008490
      INUM = 0                     00008500
      ILDDRT = LDDRTS              00008510
C                                          00008520
C*****                                00008530
C* COMPUTE THE CHANGE IN DEPOT REPAIR FRACTION FOR THE NUMBER OF *00008540
C* DERIVATIVES REQUESTED. *00008550
C*****                                00008560
C                                          00008570
      DO 19000 I=1,34              00008580
      IF (ILDDRT.GT.6) GO TO 18100 00008590
      INUM = ILDDRT + INUM         00008600
      IPIN = 1                     00008610
      GO TO 18150                  00008620
18100  INUM = 6 + INUM            00008630
      ILDDRT = ILDDRT-6           00008640
C                                          00008650
C*****                                00008660
C* PRINT OUT DERIVATIVES AT TERMINAL (6 PER LINE) *00008670
C*****                                00008680
C                                          00008690
18150  WRITE (6,4) (IDDRTS(J),J=IBEG,INUM) 00008700
      WRITE (6,36) (TEMP(J),J=IBEG,INUM) 00008710
      WRITE (6,5) (TEMP2(J),J=IBEG,INUM) 00008720
      IBEG = INUM + 1              00008730
C                                          00008740
      IF (IPIN.EQ.1) GO TO 19001 00008750
19000 CONTINUE                    00008760
C                                          00008770
19001 IF (PRNT.EQ.0) GO TO 20001 00008780
      WRITE (7,35)                 00008790
      IPIN = 0                     00008800

```

```

IBEG = 1
INUM = 0
ILDDRT = MAXO(LDERV,LDDRTS)
C
DO 20000 I=1,17
C
IF (ILDDRT.GT.12) GO TO 19100
INUM = ILDDRT + INUM
IPIN = 1
GO TO 19150
19100 INUM = 12 + INUM
ILDDRT = ILDDRT-12
C
C*****
C* PRINT OUT DERIVATIVES AT THE OFF-LINE PRINTER (12 PER LINE) *
C*****
C
19150 WRITE (7,6) (IDDRTS(J),J=IBEG,INUM)
WRITE (7,37) (TEMP(J),J=IBEG,INUM)
WRITE (7,7) (TEMP2(J),J=IBEG,INUM)
IBEG = INUM + 1
C
IF (IPIN.EQ.1) GO TO 20001
20000 CONTINUE
C
C*****
C* THE ELEVENTH SECTION PRINTS OUT LRU MISSION CRITICAL ORDERED BY *
C* SENSITIVITY FOR LRUS WITH MCI >= 1. *
C*****
C
20001 IF (LDMCI.EQ.0.AND.LDERV.EQ.0) GO TO 22001
C
ILDMCI = MAXO(LDERV,LDMCI)
DO 20050 I=1,ILDMCI
ITEMP(I) = MCIC(IDMCI(I))
TEMP2(I) = TDMCI(IDMCI(I)) / 1000000.
20050 CONTINUE
C
IF (LDMCI.EQ.0.OR.PRNT.EQ.1) GO TO 21001
WRITE (6,38)
IPIN = 0
IBEG = 1
INUM = 0
ILDMCI = LDMCI
C
C*****
C* COMPUTE THE CHANGE IN MCI FOR NUMBER OF DERIVATIVES REQUESTED. *
C*****
C
DO 21000 I=1,34
IF (ILDMCI.GT.6) GO TO 20100
INUM = ILDMCI + INUM
IPIN = 1
GO TO 20150
20100 INUM = 6 + INUM
ILDMCI = ILDMCI-6
C
C*****
C* PRINT OUT DERIVATIVES AT TERMINAL (6 PER LINE) *
C*****
C
20150 WRITE (6,4) (IDMCI(J),J=IBEG,INUM)
WRITE (6,39) (ITEMP(J),J=IBEG,INUM)

```

WRITE (6,5) (TEMP2(J),J=IBEG,INUM)	00009440
IBEG = INUM + 1	00009450
C	00009460
IF (IPIN.EQ.1) GO TO 21001	00009470
21000 CONTINUE	00009480
C	00009490
21001 IF (PRNT.EQ.0) GO TO 22001	00009500
WRITE (7,38)	00009510
IPIN = 0	00009520
IBEG = 1	00009530
INUM = 0	00009540
ILDNCI = MAX0(LDERV,LDNCI)	00009550
C	00009560
DO 22000 I=1,17	00009570
C	00009580
IF (ILDNCI.GT.12) GO TO 21100	00009590
INUM = ILDNCI + INUM	00009600
IPIN = 1	00009610
GO TO 21150	00009620
21100 INUM = 12 + INUM	00009630
ILDNCI = ILDNCI-12	00009640
C	00009650
C*****	00009660
C* PRINT OUT DERIVATIVES AT THE OFF-LINE PRINTER (12 PER LINE)	*00009670
C*****	00009680
C	00009690
21150 WRITE (7,6) (IDNCI(J),J=IBEG,INUM)	00009700
WRITE (7,40) (ITEMP(J),J=IBEG,INUM)	00009710
WRITE (7,7) (TEMP2(J),J=IBEG,INUM)	00009720
IBEG = INUM + 1	00009730
C	00009740
IF (IPIN.EQ.1) GO TO 22001	00009750
22000 CONTINUE	00009760
C	00009770
22001 RETURN	00009780
END	00009790


```

SUBROUTINE APRINT                                00000010
C                                                    00000020
C*****00000030
C                                                    00000040
C*****00000050
C*          COMMON BLOCK ORGANIZATION                00000060
C* IN GENERAL, FOR EVERY SUBROUTINE THERE IS A COMMON BLOCK THAT 00000070
C* CONTAINS ALL OF THE VARIABLES THAT FIRST RECEIVED VALUES IN THAT 00000080
C* SUBROUTINE. HENCE, ALL VARIABLES THAT ARE READ FROM THE UNIT 11 00000090
C* FILE IN SUBROUTINE READ1 WILL BE IN COMMON /RD1/; ALL VARIABLES 00000100
C* CALCULATED IN SUBROUTINE AUXIL1 WILL BE IN COMMON /AUX1/; ALL 00000110
C* VARIABLES CALCULATED IN SUBROUTINE COST1 WILL BE IN COMMON /C1/; 00000120
C* ALL VARIABLES CALCULATED IN SUBROUTINE DPR WILL BE IN COMMON /TDPR/. 00000130
C*****00000140
C                                                    00000150
C      COMMON /CNTL/ EXIT,ITER,MAXPNT,PRNT,REDO,REREAD 00000160
C      INTEGER EXIT,PRNT,REDO,REREAD 00000170
C                                                    00000180
C      COMMON /ERROR/ IERROR,IWARN 00000190
C                                                    00000200
C      COMMON /INIT/ CONFLO,MAXFA,MAXLR,MAXSE 00000210
C                                                    00000220
C      COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCMP,CDR,CDWH,CLR,CONF,CPMI, 00000230
C      * CPMP,CPWT,CRCT,DAA,DATA,DLR,DRCT,DS, 00000240
C      * FAC(10),FCS,H,INC,K,H,MCRS,MAXHRS, 00000250
C      * NCP,NDP,NSP,OSA,OST,PIUP,PNE,PPRS,REFURB,RMC, 00000260
C      * SA,SAA,SHTBI,SOSI,SPH,SPRS,SRCT,STE,SWDEV,SWFAC, 00000270
C      * TC,TCCAD,TE,TR,TW,TWCAD,XUC,YOH,YSCAD,YSLR 00000280
C      INTEGER CADRE 00000290
C      REAL INC,K,MCRS,MAXHRS 00000300
C                                                    00000310
C      COMMON /RD2/ A,COC(150),COD(150),COS(150),INOSE(150),NSEC(150), 00000320
C      * NSED(150),NSES(150),SEC(150),SENAME(150,24) 00000330
C      REAL NSEC,NSED,NSES 00000340
C      INTEGER A 00000350
C                                                    00000360
C      COMMON /RD3/ ENYR(10),PA,PANAME(10,24),HWCF(10),INOFA(10),MMH(10), 00000370
C      * SIZE(10),SHI(10),SWCF(10),SWFIX(10),SWVAR(10), 00000380
C      * WEIGHT(10) 00000390
C      INTEGER PA 00000400
C      REAL MMH 00000410
C                                                    00000420
C      COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),HCI(200), 00000430
C      * MTBI(200),N,PA(200),QPA(200),QR(200),RH(200), 00000440
C      * UC(200) 00000450
C      INTEGER QPA,QR 00000460
C      REAL MTBI 00000470
C                                                    00000480
C      COMMON /RD5/ BCNH(200),CBCNH(200),CMH(200),CRTS1(200),CRTS2(200), 00000490
C      * DBCNH(200),DHH(200),DRTS1(200),DRTS2(200), 00000500
C      * DRTS3(200),FPR(200),IMH(200),INO15(200),N15, 00000510
C      * PANH(200),RIP(200),RL(200),RMH(200),SHH(200),SRTS1(200), 00000520
C      * WOR1(200),WOR2(200),WOR3(200) 00000530
C      REAL IMH 00000540
C      INTEGER RL 00000550
C                                                    00000560
C      COMMON /SENS/ FINC,LDCNH,LDCRTS,LDDNH,LDDRTS,LDERV,LDFPR,LDFR, 00000570
C      * LDHCI,LDRH,LDSRTS,LDWOR,LDOC 00000580
C                                                    00000590
C      COMMON /AUX1/ CCOND(200),COND(200),CRTS(200),DCOND(200), 00000600
C      * DRTS(200),SNRTS(200),SRTS(200),WR(200) 00000610
C                                                    00000620
C      COMMON /AUX2/ YPR(200),WPR(200) 00000630

```

C	COMMON /AUX3/ CCLH(200),DCLH(200),DLH	00000640
C		00000650
C	COMMON /AUX4/ CAS(200),DAS(200),SAS(200)	00000660
C		00000670
C	COMMON /AUX5/ DSTK(200),ECMT,HCF(200),STK(200),STK1(200),Z	00000680
	INTEGER STK,STK1	00000690
C		00000700
C	COMMON /C1/ C1,C1D,C1P	00000710
C		00000720
C	COMMON /C2/ C2,C2C,C2D,C2L(200),C2S	00000730
C		00000740
C	COMMON /C3/ C3,C3C,C3D,C3L(200),C3S	00000750
C		00000760
C	COMMON /C4/ C4,C4I,C4L(200),C4R	00000770
C		00000780
C	COMMON /C5/ C5	00000790
C		00000800
C	COMMON /C6/ C6,C6C,C6D,C6I,C6R,C6S	00000810
C		00000820
C	COMMON /C7/ C7	00000830
C		00000840
C	COMMON /C8/ C8,C8C,C8D,C8I,C8R,C8S	00000850
C		00000860
C	COMMON /C9/ C9,C9C,C9D,C9I,C9L(200),C9R,C9S	00000870
C		00000880
C	COMMON /C10/ C10,C10HW(10),C10I,C10R,C10SW(10)	00000890
C		00000900
C	COMMON /NCOS/ CLH,PNFA(10),SMH,SPH,STH	00000910
C		00000920
C	COMMON /TDXUC/ TDXUC	00000930
C		00000940
C	COMMON /TDFR/ IDFR(200),TDFR(200)	00000950
C		00000960
C	COMMON /TDK/ TDK	00000970
C		00000980
C	COMMON /TDUC/ IDUC(200),TDUC(200)	00000990
C		00010000
C	COMMON /TDWOR/ IDWOR(200),TDWOR(200),WF(200)	00010100
C		00010200
C	COMMON /TDFPR/ IDFPR(200),TDFPR(200)	00010300
C		00010400
C	COMMON /TDCMH/ IDC MH(200),TDCMH(200)	00010500
C		00010600
C	COMMON /TDDMH/ IDDMH(200),TDDMH(200)	00010700
C		00010800
C	COMMON /TDRH/ IDR H(200),TDRH(200)	00010900
C		00011000
C	COMMON /TDSRTS/ IDSRTS(200),SFSRTS(200),TDSRTS(200)	00011100
C		00011200
C	COMMON /TDCRTS/ CFCRTS(200),IDCRTS(200),TDCRTS(200)	00011300
C		00011400
C	COMMON /TDDRTS/ DFDRTS(200),IDDRTS(200),TDDRTS(200)	00011500
C		00011600
C	COMMON /TDMCI/ IDMCI(200),MCIC(200),TDMCI(200)	00011700
C		00011800
C	COMMON /CHLCC/ CCF,CF,CHLCC,DCF,DF,SCF,SF	00011900
C		00012000
C		00012100
C		00012200
1	FORMAT ('1 F1NC=',F6.3,' LDCNH=',I3,' LDCRTS=',I3,	00012300
*	' LDDMH=',I3,' LDDRTS=',I3/' LDERV=',I3,' LDFPR=',I3,	00012400
*	' LDPR=',I3,' LDMCI=',I3,' LDRH=',I3,' LDSRTS=',I3/	00012500
*	' LDWOR=',I3,' LDUC=',I3/)	00012600

```

2  FORMAT (1X,' EXIT=',I3,' ITER=',I3,' MAXPMT=',I3/      00001270
+ ' PRNT=',I3,' REREAD=',I3/)      00001280
3  FORMAT (1X,' CONFLO=',F8.5,' MAXPA=',I3,' MAXLR=',I3,      00001290
+ ' MAXSE=',I3/)      00001300
4  FORMAT (1X,' I=',I3,' CCOND=',F6.3,' COND=',F6.3,      00001310
+ ' CRTS=',F6.3,' DCOND=',F6.3)      00001320
5  FORMAT (1X,' I=',I3,' DRTS=',F6.3,' SNRTS=',F6.3,      00001330
+ ' SRTS=',F6.3,' WR=',F6.3)      00001340
6  FORMAT (1X)      00001350
7  FORMAT (1X,' I=',I3,' YPR=',F12.3,' WPR=',F12.3)      00001360
8  FORMAT (1X,' I=',I3,' CCLH=',F12.3,' DCLH=',F12.3)      00001370
9  FORMAT (1X,' DLH=',F12.3)      00001380
10 FORMAT (1X,' I=',I3,' CAS=',F12.3,' DAS=',F12.3,' SAS=',F12.3) 00001390
11 FORMAT (1X,' I=',I3,' DSTK=',F12.5,' MCF=',I5,' STK=',I5,      00001400
+ ' STK1=',I5)      00001410
12 FORMAT (1X/' ECMT=',F12.5)      00001420
13 FORMAT (' I=',I3,' C2L=',E10.5,' C3L=',E10.5,' C4L=',E10.5,      00001430
+ ' C9L=',E10.5)      00001435
14 FORMAT (' C1D=',E10.5,' C1P=',E10.5,' C1=',E10.5/      00001440
+ ' C2C=',E10.5,' C2D=',E10.5,' C2S=',E10.5,' C2=',E10.5/      00001450
+ ' C3C=',E10.5,' C3D=',E10.5,' C3S=',E10.5,' C3=',E10.5/      00001460
+ ' C4I=',E10.5,' C4R=',E10.5,' C4=',E10.5,' C5=',E10.5/      00001470
+ ' C6C=',E10.5,' C6D=',E10.5,' C6S=',E10.5,' C6I=',E10.5,' C6R=',      00001480
+ ' E10.5,' C6=',E10.5,' C7=',E10.5/      00001490
+ ' C8C=',E10.5,' C8D=',E10.5,' C8S=',E10.5,' C8I=',E10.5,' C8R=',      00001500
+ ' E10.5,' C8=',E10.5/      00001510
+ ' C9C=',E10.5,' C9D=',E10.5,' C9S=',E10.5,' C9I=',E10.5,' C9R=',      00001520
+ ' E10.5,' C9=',E10.5,' C10I=',E10.5,' C10R=',E10.5,' C10=',E10.5/      00001530
+ ' TOTAL LCC COST=',E14.9/      00001535
+ ' CLH=',E10.5,' SMNH=',E10.5,' SPNH=',E10.5,' STMH=',E10.5)      00001540
15 FORMAT (1X/' Z=',E10.5)      00001550
C      00001560
C      00001570
LL=7      00001580
IF (PRNT.EQ.0) LL=6      00001590
WRITE (LL,1) FINC,LDCMH,LDCRTS,LDDMH,LDDRTS,LDERV,LDFPR,LDFR,      00001600
+ LDHCI,LDRH,LDSRTS,LDWOR,LDUC      00001610
WRITE (LL,2) EXIT,ITER,MAXPMT,PRNT,REREAD      00001620
WRITE (LL,3) CONFLO,MAXPA,MAXLR,MAXSE      00001630
C      00001640
DO 55 I=1,N      00001650
WRITE (LL,4) I,CCOND(I),COND(I),CRTS(I),DCOND(I)      00001660
55 CONTINUE      00001670
DO 56 I=1,N      00001680
WRITE (LL,5) I,DRTS(I),SNRTS(I),SRTS(I),WR(I)      00001690
56 CONTINUE      00001700
C      00001710
WRITE (LL,6)      00001720
DO 57 I=1,N      00001730
WRITE (LL,7) I,YPR(I),WPR(I)      00001740
57 CONTINUE      00001750
C      00001760
WRITE (LL,6)      00001770
DO 58 I=1,N      00001780
WRITE (LL,8) I,CCLH(I),DCLH(I)      00001790
58 CONTINUE      00001800
WRITE (LL,9) DLH      00001810
C      00001820
WRITE (LL,6)      00001830
DO 59 I=1,N      00001840
WRITE (LL,10) I,CAS(I),DAS(I),SAS(I)      00001850
59 CONTINUE      00001860
C      00001870

```


	WRITE(LL,6)	00001880
	DO 60 I=1,N	00001890
	WRITE(LL,11) I,DSTK(I),MCF(I),STK(I),STK1(I)	00001900
60	CONTINUE	00001910
	WRITE(LL,12) ECMT	00001920
	WRITE(LL,6)	00001930
	DO 61 I=1,N	00001940
	WRITE(LL,13) I,C2L(I),C3L(I),C4L(I),C9L(I)	00001950
61	CONTINUE	00001960
C		00001970
	CTOT = C1+C2+C3+C4+C5+C6+C7+C8+C9+C10	00001975
	WRITE(LL,14) C1D,C1P,C1,C2C,C2D,C2S,C2,	00001980
	+ C3C,C3D,C3S,C3,C4I,C4R,C4,C5,C6C,C6D,C6S,	00001985
	+ C6I,C6R,C6,C7,C8C,C8D,C8S,C8I,C8R,C8,C9C,C9D,C9S,C9I,C9R,C9,	00001990
	+ C10I,C10R,C10,CTOT,CLH,SNHH,SPNH,STNH	00002000
C		00002010
	WRITE(LL,15) Z	00002020
C		00002030
	RETURN	00002040
	END	00002050


```

SUBROUTINE DPRINT
C
C*****00000010
C*****00000020
C*****00000030
C*      SEEK IGLOO LIFE CYCLE COST MODEL      *00000040
C* THIS PROGRAM IS DESIGNED TO RUN REPETITIVE LIFE CYCLE COST *00000050
C* CALCULATIONS IN AN INTERACTIVE ENVIRONMENT.  THERE ARE SIX INPUT *00000060
C* DATA SETS:                                *00000070
C*   UNIT 5 - RECEIVES INPUTS TYPED BY THE USER AT THE TERMINAL. *00000080
C*   UNIT 11 - MISCELLANEOUS INPUTS (MAINLY SCALARS). *00000090
C*   UNIT 12 - INPUTS BY SUPPORT EQUIPMENT TYPE. *00000100
C*   UNIT 13 - INPUTS BY FUNCTIONAL AREA. *00000110
C*   UNIT 14 - INPUTS BY LRU TYPE (LRU FILE 1). *00000120
C*   UNIT 15 - INPUTS BY LRU TYPE (LRU FILE 2). *00000130
C* TWO FILES RECEIVE THE OUTPUT: *00000140
C*   UNIT 6 - OUTPUT THAT GOES TO THE TERMINAL. *00000150
C*   UNIT 7 - OUTPUT THAT GOES TO THE OFF-LINE PRINTER. *00000160
C*****00000170
C*****00000180
C*****00000190
C*      COMMON BLOCK ORGANIZATION *00000200
C* IN GENERAL, FOR EVERY SUBROUTINE THERE IS A COMMON BLOCK THAT *00000210
C* CONTAINS ALL OF THE VARIABLES THAT FIRST RECEIVED VALUES IN THAT *00000220
C* SUBROUTINE.  HENCE, ALL VARIABLES THAT ARE READ FROM THE UNIT 11 *00000230
C* FILE IN SUBROUTINE READ1 WILL BE IN COMMON /RD1/; ALL VARIABLES *00000240
C* CALCULATED IN SUBROUTINE AUXIL1 WILL BE IN COMMON /AUX1/; ALL *00000250
C* VARIABLES CALCULATED IN SUBROUTINE COST1 WILL BE IN COMMON /C1/; *00000260
C* ALL VARIABLES CALCULATED IN SUBROUTINE DPR WILL BE IN COMMON /DPR/. *00000270
C*****00000280
C      COMMON /CNTL/ EXIT,ITER,MAXPMT,PRMT,REDO,BEREAD *00000290
C      INTEGER EXIT,PRMT,REDO,REREAD *00000300
C      COMMON /ERROR/ IERROR,IWARN *00000310
C      COMMON /INIT/ COMFLO,MAXPA,MAXLR,MAXSE *00000320
C      COMMON /RD1/ ADCH,ADPH,B,CAA,CADRE,CCHP,CDR,CDWH,CLR,CONF,CPNI, *00000330
C      + CPNP,CPWT,CRCT,DAA,DATA,DLR,DRCT,DS, *00000340
C      + PAC(10),PCS,H,INC,K,N,MCRS,MXHS, *00000350
C      + NCP,NDP,NSP,OSA,OST,PIUP,PNE,PPRS,REFURB,RMC, *00000360
C      + SA,SAA,SMTBI,SOST,SPN,SPRS,SRCT,STE,SWDEV,SWFAC, *00000370
C      + TC,TCCAD,TE,TR,TW,TWCAD,XUC,YOH,YSCAD,YSLR *00000380
C      INTEGER CADRE *00000390
C      REAL IMC,K,MCRS,MXHS *00000400
C      COMMON /RD2/ A,COC(150),COD(150),COS(150),INOSE(150),NSEC(150), *00000410
C      + NSED(150),NSES(150),SEC(150),SENAME(150,24) *00000420
C      REAL NSEC,NSED,NSES *00000430
C      INTEGER A *00000440
C      COMMON /RD3/ ENYR(10),PA,PANAME(10,24),HWCF(10),INOFA(10),MMH(10), *00000450
C      + SIZE(10),SMI(10),SWCF(10),SWFIX(10),SWVAR(10), *00000460
C      + WEIGHT(10) *00000470
C      INTEGER PA *00000480
C      REAL MMH *00000490
C      COMMON /RD4/ INO(200),LRNAME(200,24),LRPART(200,12),MCI(200), *00000500
C      + MTBI(200),N,PA(200),QPA(200),QR(200),RH(200), *00000510
C      + UC(200) *00000520
C      INTEGER QPA,QR *00000530
C      REAL MTBI *00000540
C      COMMON /RD5/ BCMH(200),CBCMH(200),CHH(200),CRTS1(200),CRTS2(200), *00000550
C      + *00000560
C      + *00000570
C      + *00000580
C      + *00000590
C      + *00000600
C      + *00000610
C      + *00000620
C      + *00000630

```

	+	DBCMM (200) , DMM (200) , DRTS1 (200) , DRTS2 (200) ,	00000640
	+	DRTS3 (200) , FPR (200) , INH (200) , INO15 (200) , N15 ,	00000650
	+	PANH (200) , RIP (200) , RL (200) , RMH (200) , SMH (200) , SRTS1 (200) ,	00000660
	+	WOR1 (200) , WOR2 (200) , WOR3 (200)	00000670
		REAL INH	00000680
		INTEGER RL	00000690
C		COMMON /SENS/ FINC, LDCMH, LDCRTS, LDDMH, LDDRTS, LDERV, LDFPR, LDFR,	00000700
	+	LDHCL, LDRH, LDSRTS, LDWOR, LDUC	00000710
C		COMMON /AUX1/ CCOND (200) , COND (200) , CRTS (200) , DCOND (200) ,	00000720
	+	DRTS (200) , SMRTS (200) , SRTS (200) , WR (200)	00000730
C		COMMON /AUX2/ YPR (200) , WPR (200)	00000740
C		COMMON /AUX3/ CCLH (200) , DCLH (200) , DLH	00000750
C		COMMON /AUX4/ CAS (200) , DAS (200) , SAS (200)	00000760
C		COMMON /AUX5/ DSTK (200) , ECMT, MCF (200) , STK (200) , STK1 (200) , Z	00000770
		INTEGER STK, STK1	00000780
C		COMMON /C1/ C1, C1D, C1P	00000790
C		COMMON /C2/ C2, C2C, C2D, C2L (200) , C2S	00000800
C		COMMON /C3/ C3, C3C, C3D, C3L (200) , C3S	00000810
C		COMMON /C4/ C4, C4I, C4L (200) , C4R	00000820
C		COMMON /C5/ C5	00000830
C		COMMON /C6/ C6, C6C, C6D, C6I, C6R, C6S	00000840
C		COMMON /C7/ C7	00000850
C		COMMON /C8/ C8, C8C, C8D, C8I, C8R, C8S	00000860
C		COMMON /C9/ C9, C9C, C9D, C9I, C9L (200) , C9R, C9S	00000870
C		COMMON /C10/ C10, C10HW (10) , C10I, C10R, C10SW (10)	00000880
C		COMMON /WCOS/ CLH, PHFA (10) , SMH, SPH, STH	00000890
C		COMMON /TDKUC/ TDKUC	00000900
C		COMMON /TDFR/ IDFR (200) , TDFR (200)	00000910
C		COMMON /TDR/ TDR	00000920
C		COMMON /TDUC/ IDUC (200) , TDUC (200)	00000930
C		COMMON /TDWOR/ IDWOR (200) , TDWOR (200) , WP (200)	00000940
C		COMMON /TDFPR/ IDFPR (200) , TDFPR (200)	00000950
C		COMMON /TDCMH/ IDCMM (200) , TDCMH (200)	00000960
C		COMMON /TDDMH/ IDDMH (200) , TDDMH (200)	00000970
C		COMMON /TDRH/ IDRH (200) , TDRH (200)	00000980
C		COMMON /TDSRTS/ IDSRTS (200) , SPSRTS (200) , TDSRTS (200)	00000990

C	COMMON /TDCRTS/ CFCRTS(200),IDCRTS(200),TDCRTS(200)	00001270
C	COMMON /TDDRTS/ DFDRTS(200),IDDRTS(200),TDDRTS(200)	00001280
C	COMMON /TDMCI/ IDHCI(200),HCIC(200),TDMCI(200)	00001290
C	COMMON /CHLCC/ CCF,CF,CHLCC,DCF,DF,SCF,SF	00001300
C		00001310
C		00001320
C		00001330
C		00001340
C		00001350
C		00001360
	DO 2 I=1,N	00001370
	WRITE(7,1) I,TDFR(I),TDUC(I),TDRN(I),TDFPR(I)	00001380
1	FORMAT (' I=',I3,' TDFR=',E11.5,' TDUC=',E11.5,' TDRN=',E11.5,	00001390
+	' TDFPR=',E11.5)	00001400
2	CONTINUE	00001410
C		00001420
	DO 4 I=1,N	00001430
	WRITE(7,3) I,TDCNH(I),TDDNH(I),TDMCI(I),TDWOR(I)	00001440
3	FORMAT (' I=',I3,' TDCNH=',E11.5,' TDDNH=',E11.5,' TDMCI=',	00001450
+	E11.5,' TDWOR=',E11.5)	00001460
4	CONTINUE	00001470
C		00001480
	DO 6 I=1,N	00001490
	WRITE(7,5) I,TDSRTS(I),TDCRTS(I),TDDRTS(I)	00001500
5	FORMAT (' I=',I3,' TDSRTS=',E11.5,' TDCRTS=',E11.5,' TDDRTS=',	00001510
+	E11.5)	00001520
6	CONTINUE	00001530
	IF (PRNT.NE.0) WRITE(7,7) TDXUC,TDK	00001540
	IF (PRNT.NE.1) WRITE(6,7) TDXUC,TDK	00001545
7	FORMAT (' DXUC=',E11.5,' DK=',E11.5)	00001550
C		00001560
	RETURN	00001570
	END	00001580